U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

PB-292 760

FEDBAK03 - A Computer Program for the Modelling of First Order Consecutive Reactions with Feedback Under a Steady State Multidimensional Natural Aquatic System Program Documentation and User's Guide

(U.S.) Environmental Protection Agency, New York. Region II

Nov 78

United States
Environmental Protection
Agency

Region 2
26 Federal Plaza
New York NY 10007

New Jersey New York
Puerto Rico
Virgin Islands

PB 292760

EPA

FEDBAKØ3 - A COMPUTER PROGRAM FOR THE MODELLING OF FIRST ORDER CONSECUTIVE REACTIONS WITH FEEDBACK UNDER A STEADY STATE MULTIDIMENSIONAL NATURAL AQUATIC SYSTEM

PROGRAM DOCUMENTATION AND USERS GUIDE

George A. Nossa
Chief, Environmental Systems Section
Data Systems Branch
Management Division
U.S. Environmental Protection Agency
New York, New York 10007
November 1978

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

BIBLIOGRAPHIC DATA SHEET	1. Report No. EPA-902/2-79-001	2.	3. Recipient's Accession No. PB292760
4. Title and Subtitle A Computer Program for the Modelling of First Order Consecutive Reactions with Feedback under a Steady State Multidimensional Natural Aquatic System			5. Report Date November 1978 6.
7. Author(s) NOSSA, GEORGE A.			8. Performing Organization Rept. No.
9. Performing Organization Name and Address United States Environmental Protection Agency, Reg II Data Systems Branch 26 Federal Plaza New York, N.Y. 10007			10. Project/Task/Work Unit No. 11. Contract/Grant No.
12. Sponsoring Organization Name and Address Same as above.			13. Type of Report & Period Covered Final 14.

15. Supplementary Notes

16. Abstracts The computer model described is used to compute the steady-state distribution of water quality variables undergoing consecutive reactions with feedback and following first order kinetics. The program has been developed in a general form but is specifically applicable to the reactions observed by nitrogenous species and the associated dissolved oxygen uptake in the natural environment.

17. Key Words and Document Analysis. 17a. Descriptors

*Mathematical Model, *Water Quality, *Computer Model, Simulation Analysis, Oxygen Demand, Nutrients.

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. Pages
Release to Public	UNCLASSIFIED	93
	20. Security Class (This Page) UNCLASSIFIED	22. Price PC A05 MA01

FORM NTIS-35 (REV. 10-73) ENDORSED BY ANSI AND UNESCO. THIS FORM MAY BE REPRODUCED USCOMM-DC 8265-P74

TABLE OF CONTENTS

Abstract - 2
Introduction - 2
Theory - 2
Acknowledgements - 6
References - 6

Description of the Main Program and Accompanying subroutines - - - - - - - 7
Program Flowchart - 8
Source Program Listing - 16
Input Description - 38
Restrictions - 49

Application to the Delaware Estuary - 50
Description of Area of Study - 51
Nitrogenous Point Source Loadings - - - - - - - - - - - - - - - - - 53
Program Output - 54
Graphs of Computed vs Observed Values - - - - - - - - - - - - - - 72

Program Execution at the COMNET facility - - - - - - - - - - - - - - - - - 83

Acknowledgements - 6,84

References to the Computer Program - 85

APPENDIX A
Listing of Input data for Delaware Estuary Verification - - - 86

FEDBAKØ3 - A COMPUTER PROGRAM FOR THE MODELLING OF FIRST ORDER CONSECUTIVE REACTIONS WITH FEEDBACK UNDER A STEADY STATE MULTIDIMENSIONAL NATURAL AQUATIC SYSTEM *

George A. Nossa
Environmental Engineer
Data Systems Branch
U.S. Environmental Protection Agency
New York, N.Y.

ABSTRACT

The computer model described is used to compute the steady-state distribution of water quality variables undergoing consecutive reactions with feedback and following first order kinetics. The program has been developed in a general form but is specifically applicable to the reactions observed by nitrogenous species and the associated dissolved oxygen uptake in the natural environment.

The basis for this model is the theory of conservation of mass. The approach used to solve the equations is a finite difference scheme developed by Thomann [4,5], which has been shown to be a very effective tool in the field of water quality management.

INTRODUCTION

A computer model has been developed to serve as a useful tool in the prediction of water quality parameters which react under first order kinetics and as a system of consecutive reactions,where any parameter can react in a feedback fashion. The problem setting assumes an aquatic environment in which steady state conditions can be applied

Lets consider a system of five reactants:

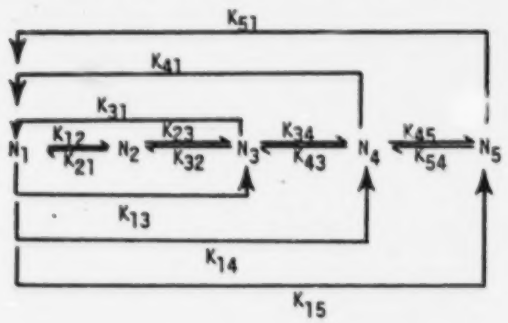


FIGURE 1
SCHEMATIC OF FIVE REACTANT SYSTEM

In the above system, the consecutive feedfoward and feedback reaction rates are presented. All the possible reaction loops for the first reactant have also been included. This particular system will be used in the theory development.

THEORY

The general estuarine advection/dispersion equation may be written as:

$$\frac{\delta N}{\delta t} = \frac{E\delta^2 N}{\delta x^2} - \frac{U\delta N}{\delta x} - KN + W_N(x) \quad \ldots\ldots\ldots\ldots (1)$$

where:

N = concentration of constituent
t = time, in tidal cycles
E = tidally averaged dispersion coefficient which includes the dispersive effects of tidal motion
U = net advective velocity
K = first order decay coefficient of constituent N
W(x) = direct discharges of N

Assuming steady state conditions: dN/dt = 0 and equation (1) becomes:

$$0 = E\frac{d^2N}{dx^2} - U\frac{dN}{dx} - KN + W(x) \quad \ldots\ldots\ldots\ldots (2)$$

Direct solutions for the above equation have been computed by O'Connor[1,2], a computer program has been documented by EPA, Region II[3] which uses this technique to solve for water quality parameters.

A second solution approach developed by Thomann[4,5] solves the above differential equation directly by replacing the derivatives with finite-difference approximations. This approach is used by computer program HARØ3; documented by EPA, Region II[6] to analyze systems of consecutive reactions. The Thomann solution technique will also be the approach followed in program FEDBAKØ3.

If we take the first reactant N_1 on figure 1 and we incorporate all the feedback loops, equation (2) becomes:

$$0 = E\frac{d^2N_1}{dx^2} - U\frac{dN_1}{dx} - K_{11}N_1+K_{21}N_2+K_{31}N_3+K_{41}N_4+K_{51}N_5+W_1 \quad \ldots\ldots\ldots (3)$$

where K_{ij} are the appropriate first order reaction rate constants. The incorporation of the reaction term K_{11} allows for the first order decay of this first component out of the system.

For the other remaining reactants equation (2) would be written as:

$$0 = E\frac{d^2N_2}{dx^2} - U\frac{dN_2}{dx} - K_{22}N_2+K_{12}N_2+K_{32}N_3+K_{42}N_4+K_{52}N_5+W_2 \quad \ldots\ldots\ldots (4)$$

$$0 = E\frac{d^2N_3}{dx^2} - U\frac{dN_3}{dx} - K_{33}N_3+K_{13}N_1+K_{23}N_2+K_{43}N_4+K_{53}N_5+W_3 \quad \ldots\ldots\ldots (5)$$

* Reproduced from: Environmental Modelling and Simulation, USEPA Office of Research and Development, Washington, D.C., July 1976 (EPA600/9-76-016)

$$0 = E\frac{d^2N_4}{dx^2} - U\frac{dN_4}{dx} - K_{44}N_4+K_{14}N_1+K_{24}N_2+K_{34}N_3+K_{54}N_5+W_4 \quad \ldots\ldots\ldots\ldots\ldots (6)$$

$$0 = E\frac{d^2N_5}{dx^2} - U\frac{dN_5}{dx} - K_{55}N_5+K_{15}N_1+K_{25}N_2+K_{35}N_3+K_{45}N_4+W_5 \quad \ldots\ldots\ldots\ldots\ldots (7)$$

The Thomann solution approach divides the system into completely mixed segments, as illustrated for a one dimensional estuary in figure 2.

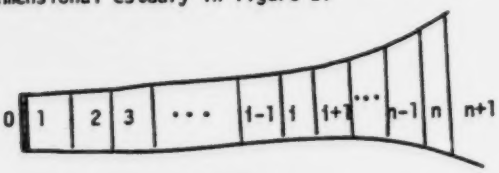


FIGURE 2
SEGMENTED ONE DIMENSIONAL ESTUARY

where segments 1 and 'n' each form an interface with the boundaries. Equation (3) for the "i" segment on figure (2) can be written as:

$$E_iA_il_i\frac{d^2N_{1,i}}{dx^2} - Q_il_i\frac{dN_{1,i}}{dx} - V_iK_{11,i}N_{1,i}+V_iK_{21,i}N_{2,i} + V_iK_{31,i}N_3+V_iK_{41,i}N_4+V_iK_{51,i}N_5+W_i=0 \quad \ldots\ldots\ldots (8)$$

The derivatives on the equation above can be replaced by finite-difference approximations giving:

$$E_iA_il_i\frac{d^2N_{1,i}}{dx^2} = E'_{i-1,i}(N_{1,i-1}-N_{1,i})+E'_{i,i+1}(N_{1,i+1}-N_{1,i}) \quad \ldots\ldots\ldots\ldots\ldots (9A)$$

$$Q_il_i\frac{dN_{1,i}}{dx} = Q_{i,i+1}(\alpha_{i,i+1}N_{1,i}+\beta_{i,i+1}N_{1,i+1}) - Q_{i-1,i}(\alpha_{i-1,i}N_{1,i-1}+\beta_{i-1,i}N_{1,i}) \quad (9B)$$

where: $E'_{i,j} = \frac{E_{i,j}A_{i,j}}{L_{i,j}}$ ..... (9C)

i,j = subscripts used to denote the interface between adjacent segments i and j

$L_{i,j}$ = average length of segments i and $j = (l_i+l_j/2)$

$\alpha_{i,j}$ and $\beta_{i,j}$ are weight factors to correct the concentrations from equation 9B

$$\alpha_{i,j} = \frac{l_j}{l_i+l_j} \quad \ldots\ldots\ldots\ldots\ldots (10)$$

$$\beta_{i,j} = 1 - \alpha_{i,j} = \frac{l_i}{l_i+l_j} \quad \ldots\ldots\ldots\ldots\ldots (11)$$

In order for the final concentrations to be positive, it is required that:

$$\alpha_{i,j} > 1 - E'_{i,j}/Q_{i,j} \quad \ldots\ldots\ldots\ldots\ldots (12)$$

Substituting equations (9A) and (9B) into equation (8) yields:

$$E'_{i-1,i}(N_{1,i-1}-N_{1,i})+E'_{i,i+1}(N_{1,i+1}-N_{1,i}) - Q_{i,i+1}(\alpha_{i,i+1}N_{1,i}+\beta_{i,i+1}N_{1,i+1}) + Q_{i-1,i}(\alpha_{i-1,i}N_{1,i-1}+\beta_{i-1,i}N_{1,i}) - V_iK_{11,i}N_{1,i} + V_iK_{21,i}N_{2,i}+V_iK_{31,i}N_{3,i}+V_iK_{41,i}+V_iK_{51,i} N_{5,i}+W_{1,i}=0 \quad \ldots\ldots\ldots\ldots\ldots (13)$$

Grouping terms in the above equation yields:

$$(-Q_{i-1,i}\alpha_{i-1,i}-E'_{i-1,i})N_{1,i-1} + (Q_{i,i+1}\alpha_{i,i+1}-Q_{i-1,i}\beta_{i-1,i}+E'_{i-1,i}+E'_{i,i+1}+ V_iK_{11,i})N_{1,i} + (Q_{i,i+1}\beta_{i,i+1}-E'_{i,i+1})N_{1,i+1}=W_{1,i}+V_iK_{21,i}N_{2,i} + V_iK_{31,i}N_3+V_iK_{41,i}N_4+V_iK_{51,i}N_5 \quad \ldots\ldots\ldots\ldots\ldots (14)$$

Letting:

$$a_{i,i-1}=-Q_{i-1,i}\alpha_{i-1,i}-E'_{i-1,i} \quad \ldots\ldots\ldots\ldots\ldots (15)$$

$$a_{i,i}=Q_{i,i+1}\alpha_{i,i+1}-Q_{i-1,i}\beta_{i-1,i}+E'_{i-1,i}+E'_{i,i+1}+ V_iK_{11,i} \quad \ldots\ldots\ldots\ldots\ldots (16)$$

$$a_{i,i+1}=Q_{i,i+1}\beta_{i,i+1}-E'_{i,i+1} \quad \ldots\ldots\ldots\ldots\ldots (17)$$

The general equation for the ith segment becomes:

$$a_{i,i-1}N_{1,i-1}+a_{i,i}N_{1,i}+a_{i,i+1}N_{1,i+1}=W_{1,i}+V_iK_{21,i} N_{2,i}+V_iK_{31,i}N_3+V_iK_{41,i}N_4+V_iK_{51,i}N_5 \quad \ldots\ldots (18)$$

The use of this finite difference approximation scheme has a numerical dispersion, which can be approximated as (5)

$$E_{num} = UL(\alpha-1/2) \quad \ldots\ldots\ldots\ldots\ldots (18A)$$

Where E_{num} is the numerical dispersion. This is particularly important to stream applications where advective velocities may be high and this effect may lead to distorted results.

Extension to multi-dimensional analysis:

If we consider a grid of orthogonally straped sections such as the one illustrated in figure 3:

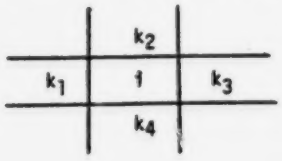


FIGURE 3
HYPOTHETICAL TWO DIMENSIONAL SYSTEM

Following the convention that flows entering a section is negative and flow out of a section is positive, a mass balance due to the transport and dispersion of material from section i to all surrounding sections k(s) is:(4,5,8)

$$V_i\frac{dN_{1,i}}{dt}=0=\sum_{\alpha}-Q_{ik}(\alpha_{ik}N_{1,i}+\beta_{ik}N_{1,k})+E'_{ik}(N_{1,k}-N_{1,i})$$
$$-V_iK_{11,i}N_{1,i}+V_iK_{21,i}N_{2,i}+\ldots+V_iK_{51,i}N_{5,i}$$
$$+W_{1,i}\quad(i=1,2\ldots n)\qquad\ldots\ldots\ldots

$$\begin{bmatrix} [A_1] & -[VK_{21}] & -[VK_{31}] & -[VK_{41}] & -[VK_{51}] \\ -[VK_{12}] & [A_2] & -[VK_{32}] & -[VK_{42}] & -[VK_{52}] \\ -[VK_{13}] & -[VK_{23}] & [A_3] & -[VK_{43}] & -[VK_{35}] \\ -[VK_{14}] & -[VK_{24}] & -[VK_{34}] & [A_4] & -[VK_{54}] \\ -[VK_{15}] & -[VK_{25}] & -[VK_{35}] & -[VK_{45}] & [A_5] \end{bmatrix} \begin{Bmatrix} (N_1) \\ (N_2) \\ (N_3) \\ (N_4) \\ (N_5) \end{Bmatrix} = \begin{Bmatrix} (W_1) \\ (W_2) \\ (W_3) \\ (W_4) \\ (W_5) \end{Bmatrix}$$

or

$$[\overline{A}]\ (\overline{N}) = (\overline{W}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (37)$$

where $[\overline{A}]$ is the 5n x 5n matrix above and $(\overline{N})$ and $(\overline{W})$ are 5n x 1 vectors. The solution of the five reactants over all the spacial sections are given by

$$(\overline{N}) = [\overline{A}]^{-1}\ (\overline{W}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (38)$$

Application of the theory by the computer program

The program described follows a modular approach in which the user specifies to the main line program the options desired, and subroutines are called accordingly to perform specific tasks.

The steps to be accomplished can be summarized as:

a) Input the physical characteristics of the system; namely, the geometry, temperature, hydrologic characteristics, reaction schemes and corresponding reaction rates.
b) Calculate E' and α's for all the sections as described on equations (9C) and (10). In order to handle the constraint stated on equation (12), the program tests the expression:

$$\alpha_{ij} < 1\text{-}E'_{ij}/Q_{ij} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (39)$$

and if such is the case α_{ij} is recalculated as:

$$\alpha_{ij} = 1\text{-}E'_{ij}/2Q_{ij} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (40)$$

which places α_{ij} well within the tolerable range.

c) Set up the system matrix $[A_i]$ by computing its elements as given on equations (22A) and (22B). It should be noted that the difference between the $[A_i]$ matrices in equations (31) thru (35) is the addition of a separate main diagonal term $K_iK_{jj,i}$. In order to conserve space, this matrix is set up without this term and during the creation of the matrix of matrices $[\overline{A}]$, the appropriate $V_iK_{jj,i}$ term is added.
d) Set up the matrix of matrices $[\overline{A}]$, this is done by combining an offline disk file containing all the V_iK_i terms for the system and the $[A_i]$ matrix.
e) Input of direct discharges into the system and boundary concentrations, and from these compute the system source vector $(\overline{W})$ as described on equations (24) and (26).
f) Solve for the reactants concentrations at all segments by inverting the matrix $[\overline{A}]$ and multiplying it by the waste vector $(\overline{W})$.

Optionally, the program can also perform system sensitivity analysis by varying the waste vector $(\overline{W})$ and re-multiplying by $[\overline{A}]^{-1}$ and/or changing the reaction rate constants for any reactants and repeating step (f). A second option is the computation of dissolved oxygen deficit and the corresponding dissolved oxygen concentration by selecting the reaction schemes producing the deficit and the associated stochiometric coefficient.

The computer program has been written for the IBM 370 with a Fortran (IV) G or H level compiler. The program occupies 140K of core to execute and takes 35 CPU seconds to solve a 10 segment, 8 component system. As presently written, the program can accommodate a multi-dimensional system of up to sixty sections and each section can have a maximum of six interfaces. The maximum number of reactants is such that when multiplied by the number of sections cannot exceed 120. This present limitation can be easily expanded.

Application to nitrification

Figure 4 is a schematic representation of the nitrogen cycle:

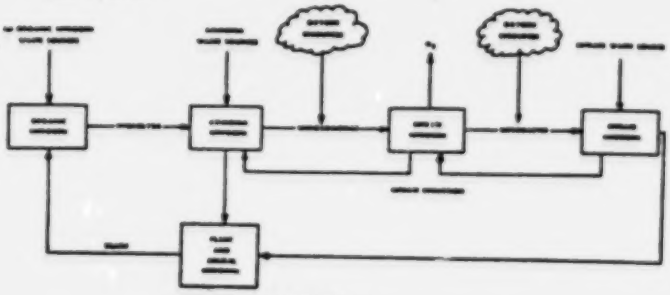


FIGURE 4
MAJOR FEATURES OF THE NITROGEN CYCLE

Since waste loads are usually in the form of organic nitrogen or ammonia, these species will consume oxygen by the bacterial reactions:(7)

$$NH_4^+ + 3/2\ O_2 \xrightarrow[\text{Bacteria}]{\text{Nitrosomas}} NO_2^- + 2H^+ + H_2O \quad \ldots\ldots\ (41)$$

followed by

$$NO_2^- + 1/2\ O_2 \xrightarrow[\text{Bacteria}]{\text{Nitrobacter}} NO_3^- \quad \ldots\ldots\ldots\ldots\ (42)$$

From the stochiometry of the reaction on equation (41), it takes 3.43 grams of oxygen for the oxydation of one gram of ammonia as nitrogen to nitrite. The second reaction takes 1.14 grams of oxygen for the oxydation of one gram of nitrite as nitrogen to nitrate. The entire oxydation process therefore takes 4.57 grams of oxygen per gram of ammonia nitrogen.

Letting:

N_1 = organic nitrogen
N_2 = ammonia nitrogen
N_3 = nitrite nitrogen
N_4 = nitrate nitrogen
N_5 = plant and animal nitrogen

The system to be solved would be:

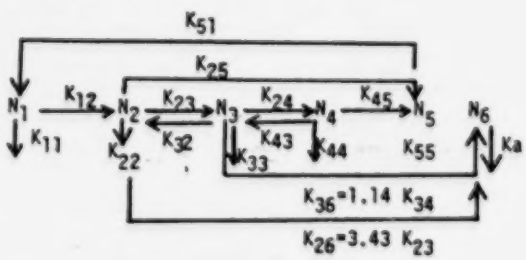


FIGURE 5
NITROGEN CYCLE WITH DEFICIT COMPONENT

If we assume these reactions to follow first order kinetics, the system can readily be solved using program FEDBAKØ3. The computation of dissolved oxygen deficit can be accomplished two ways. A deficit "species" can be defined (noted N_6 above), the decay of which is the reaeration rate, Ka. The reaction schemes producing deficit are then defined, and the corresponding reaction rate would be the product of the stochiometric coefficient by the reaction rate of the reaction using up oxygen. A second method to compute deficit concentrations as done for component N_1 in equations (22) thru (31), one obtains

$$[B](D)_1=3.43[VK_{23}](N_2)+1.14[VK_{34}](N_3) \quad \ldots\ldots \quad (43)$$

where [B] is a matrix similar to the $[A_1]$ matrices of equations (31) thru (35), except that the main diagonal term has the reaeration rate Ka instead of K_{11}, (N_2) and (N_3) are nxl vectors of the steady-state concentration of these reactants. $(D)_1$ is an nxl vector of the deficit concentrations over all segments due to the oxydation of ammonia and nitrite. The solution to the deficit concentration over all space is given by:

$$(D)_1=3.43[VK_{23}](N_2)[B]^{-1}+1.14[VK_{34}](N_3)[B]^{-1}. \quad (44)$$

This method is used to compute deficit in program FEDBAK, by using the optional subroutine.

The application to nitrification and dissolved oxygen deficit assumed first order kinetics for the bacterial reactions. This should be confirmed by laboratory studies, or the nature of the system should be carefully considered. This computer model has been found to be very useful as a predictive tool and in providing insights to the behavior of nitrogen species in the aquatic environment. On new applications this will ultimately depend on the applicatively of the underlying assumptions to the system of interest.

ACKNOWLEDGEMENTS

The author is grateful to Steve Chapra of the Great Lakes Environmental Research Laboratories and Richard Winfield of the Manhattan College Department of Environmental Engineering and Science for their review and comments on this paper.

The data for test applications of this model were made available by Dr. Richard Tortoriello of the Delaware River Basin Commission. His input to this project is gratefully acknowledged.

REFERENCES

(1) O'Connor, D.J., "Oxygen Balance of an Estuary". Jour. San. Eng. Div. ASCE, Vol 86, May 1960, pp 35-55

(2) O'Connor, D.J., "The Temporal and Spacial Distribution of Dissolved Oxygen in Streams". Water Resources Research, Vol 3, No. 1, 1967, pp 65-79.

(3) Chapra, S.C. and Gordimer S., ESOOl A Steady State, One Dimensional, Estuarine Water Quality Model, USEPA,Region II, New York, N.Y. September 1973.

(4) Thomann, R.V., "Mathematical Model for Dissolved Oxygen". Jour. San. Eng. Div., ASCE, Vol 89, No. SA5, October 1963, pp 1-30.

(5) Thomann, R.V., System Analysis and Water Quality Management, Environmental Science Division, New York, 1971.

(6) Chapra, S.C. and Nossa, G.A. HAR03 A Computer Program for the Modelling of Water Quality Parameters in Steady State Multidimensional Natural Aquatic System., USEPA, Region II, New York, N.Y. October 1974

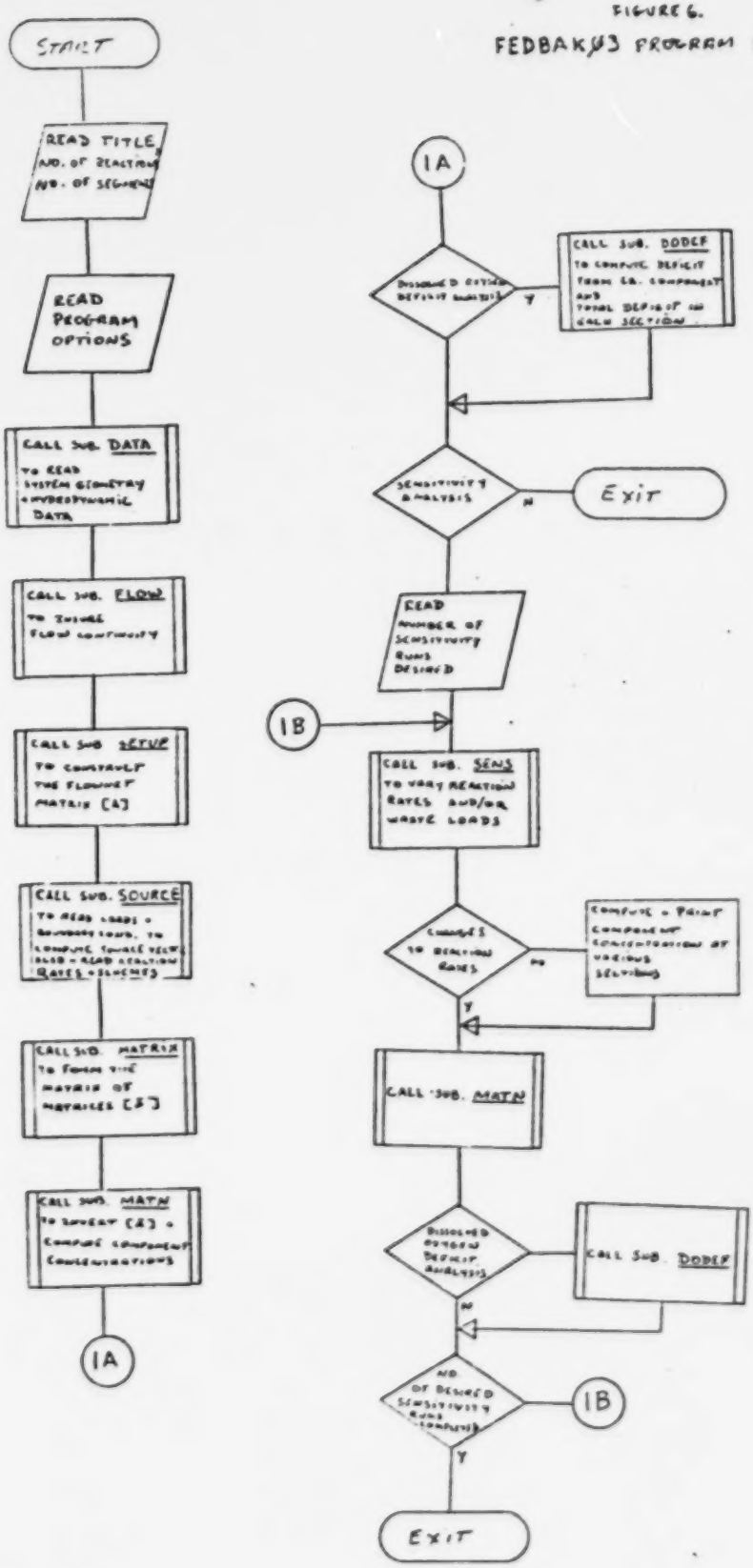
(7) Stratton, F.E. and Mc Carty, P.L., "Prediction of Nitrification Effects on the Dissolved Oxygen Balance of Streams" Env. Sci. and Tech., Vol. 1, No. 5, May 1967, pp 405-410.

(8) O'Connor, D.J., Thomann, R.V., and Di Toro, D.M., Dynamic Water Quality Forecasting and Management, USEPA, Office of Research and Development, Wash.,D.C. Report No. EPA-660/3-73-009, August 1973

DESCRIPTION OF THE MAIN PROGRAM AND ACCOMPANYING SUBROUTINES

FIGURE 6.

FEDBAK(S) PROGRAM FLOWCHART



The Main Program

FEDBAKØ3 has been programmed using a modular approach. Wherever possible, an attempt has been made to give each subroutine a unique function. The main program was designed to coordinate the various subroutines used in the model. This inter-relationship among the various subroutines is illustrated in figure 6.

Subroutine DATA...

The primary purpose of this subroutine is to input parameters which do not depend on the particular substance being modeled. These include system parameters (number of segments, indicators, etc.) and certain physical parameters such as cross-sectional area, net advective flow, etc., which are only input once when modeling a water body. These are in contrast to parameters such as loads and reaction rates which would vary with the constituent being modeled.

The particular tasks accomplished by DATA are:

1) Read and write system parameters
2) Initialize certain matrices to zero
3) Read and write interface parameters AREA, E and Q and place them into the storage matrix ARRAY.
4) Read and write the characteristic lengths (LA)
5) Read chloride concentrations (CHLOR), volume (VOL), and temperature (TEMPER) at each segment
6) Apply the following conversion factors:

$$Q(MGD) = (CFS) * .6463 \text{ MGD/cfs}$$

$$VOL(MG) = (10^6 ft^3) * 7.48 \text{ gal/ft}^3$$

At this point, a further word about the function of ARRAY is in order. ARRAY is an N X 18 sized matrix which is used to save space when storing the interface parameters AREA, E and Q. Used in conjunction with IARAY, ARRAY avoids the use of three N X N matrices when handling these variables. It does this as follows:

IARAY(I,J)	contains the section number of the section which forms the interface with section I.
ARRAY(I,J)	contains the area of interface (I,IARAY(I,J))
ARRAY(I,J + 1)	contains the E of interface (I,IARAY(I,J))
ARRAY(I,J + 2)	contains the Q of interface (I,IARAY(I,J))
J	is an integer variable which is incremented by 3 from 1 to 16

For computational purposes, every interface must be accounted for in terms of the section to which it applies. This would involve duplicate input - for instance, AREA (I,J) is the same as AREA (J,I). To simplify input, only one of the values must be input and subroutine DATA fills out the remaining positions in ARRAY.

Subroutine FLOW...

FLOW basically computes a balance of the flows into and out of a section. If there is more or less than a zero balance around the section, the resultant value is printed as an excess flow.

The purpose of this exercise is to both keep track of flows and to insure that no mistakes were made when inputting the flow regime. This can be particularly useful when working in two or three dimensions.

Subroutine SETUP...

The primary purpose of this subroutine is to set up the individual flownet matrices (A), (described in equations 31 through 35 in the theory section) and output these on a direct access device. The tasks performed by SETUP are:

1) Calculation of the bulk dispersion coefficient, E'

According to the theory:

$$E'_{i,j} = \frac{E_{i,j} A_{i,j}}{l_{i,j}}$$

Where

$E_{i,j}$ = the dispersion coefficient across the interface

$A_{i,j}$ = the cross-sectional area of the interface

$l_{i,j}$ = the average length of the adjoining sections = $(l_i + l_j)/2$

The program calculates E' (EPRIM) as follows:

```
EPRIM(J)=ARRAY(I,JJ)*ARRAY(I,JJ+1)*417.166/(LA(K,I)+LA(I,K))
```

Where

ARRAY(I,JJ) = $A_{i,j}$ (ft^2)

ARRAY(I,JJ+1) = $E_{i,j}$(mi^2/day)

LA(I,K) = l_i

LA(K,I) = l_k

To conserve computer memory these lengths are stored as a N X 6 sized array and used in conjunction with array NLA to identify the N,I interface. This again avoids the use of an N X N matrix to handle this variable.

417.1166 - conversion factor which is computed as follows:

Original Units	Conversion Factor	New Units
$\frac{ft^2 * mile^2}{day * ft * 2}$	$\frac{(5280\ ft)^2}{(mile)^2} \times \frac{7.481\ gal}{ft^3} \times 2 \times \frac{MG}{10\ gal}$	MGD

2) A value of alpha is calculated to adjust the advective term for sections of different length. This value is checked against the positivity criterion (equation 12) and if it is not met, alpha is recalculated.

```
Alpha(J) = 1.0-(EPRIM(J)/(2*ABS(ARRAY(I,JJ+2))))
```

Which corresponds to

$$\alpha = 1 - E'/2Q$$

3) The elements of the system matrix (A) are calculated and saved on mass storage

4) The values for EPRIM and ALPHA are printed

5) The section parameters: volume, temperature and chloride concentration are printed.

SUBROUTINE SOURCE

This subroutine will read the reaction schemes and corresponding reaction rate constants for each system segment and construct the (VK) system matrices. It also reads all direct discharges and boundary conditions, to form the system waste vector (W). These steps are accomplished in the following sequence:

1. The reaction coefficients are read and are corrected to the temperature in the section by the formula:

$$Ki = Ki\ \theta^{(t-20)}$$

   For the nitrification reactions, values of θ are discussed in reference (2). The corrected reaction rates are then printed.

2. The (VK) matrix is formed by multiplying by the corresponding section volumes and saved in mass storage.

3. The direct discharges of the various reactants into the system are read and used to construct the system source vector. The contribution to the source vector due to the boundary conditions are calculated from the formulas given in equations (24) and (26).

4. The system source vector is then printed out.

Subroutine MATRIX

This subroutine will:

(1) Combine the (VK) matrices on mass storage with the (A) matrices to form the system matrix of matrices ($\bar{A}$) given in equation (37)
(2) Optionally print the ($\bar{A}$) matrix
(3) Read mile points corresponding to each of the segments

Subroutine MATN

This subroutine inverts the ($\bar{A}$) matrix and multiplies it by the source vector ($\bar{W}$) giving the solution vector ($\bar{N}$). This solution vector represents the component concentrations at the various segments. The printed output includes the mile points corresponding to the beginning of each segment.

Subroutine SENS

This subroutine will perform system sensitivity analysis on the reaction rate constants and/or direct discharges into the system. The logical sequence of this subroutine is:

For Changes in direct discharges:

1.) Read and print revised loadings for section
2.) Modify original source vector to reflect the new loadings
3.) Multiply new waste vector by the inverse of system matrix
4.) Print new solution vector
5.) Optionally, the deficit analysis subroutine KBOD may be called

For Changes in reaction rates:

For changes of direct discharges and reaction rates, the first two steps above take place, the procedure below is then followed:

1.) The matrix of matrices ($\bar{A}$) is re-assembled using the ($\bar{A}$) and (VK) matrices on mass storage files.
2.) The revised reaction constant(s) at all segments in the system are read and temperature corrected.
3.) The above values are substituted in the ($\bar{A}$) matrix. (Note: Subsequent calls to this subroutine make the changes to the original ($\bar{A}$) matrix and not from the one modified in this step.). For cummulative effects this new ($\bar{A}$) matrix should be copied to the mass storage file containing the old ($\bar{A}$) matrix.
4.) Optionally print the revised ($\bar{A}$) matrix

This subroutine would then return to the main line program which in turns calls the MATN and optionally the D.O. deficit analysis routine. Then if desired it returns back to this routine for multiple runs of sensitivity analysis.

Subroutine DODEF

This subroutine will compute Dissolved Oxygen Deficit due to selected reaction schemes. For each section of the system this routine will determine the deficit due to each reactant and the resulting sum total deficit in the system. The individual steps of this routine are:

(1) Read the reaction coefficients at each section, and temperature correct these by the expression:[3]

```
KA(I)=KA(I)*1.02488(TEMPER(I)-20.)
```

(2) The temperature and reaeration rates are printed
(3) The saturation level of dissolved oxygen at each system seqment is computed from the expression.[2]

```
CS(I)=(1. -.000009*CHLOR(I))*(14.652-.41022*TEMPER(I)+
      1.0079910*TEMPER(I)**2-.000077774*TEMPER(I)**3)
```

Where:

TEMPER(I) - is the temperature of the section
CHLOR(I) - is the chloride concentration (mg/l) of the section

(4) A system deficit matrix is formed by adding the VKa term to the (A) matrix on the mass storage file (N.B. the VK term is omitted for flexibility, on the Flownet matrix (A) on mass storage)

(5) The above deficit matrix is inverted and optionally printed. The structure of the program logic requires a separate array to form the system deficit matrix and compute its inverse. In the repeated use of this subroutine during system sensitivity analysis, this inverse of the deficit matrix is saved for subsequent use.
(6) Background contributions of deficit such as benthal demand or direct sources are read into the section, the deficit source vector is formed and multiplied by the inverse of the deficit matrix giving the system deficit response due to direct loads. These results are then printed.
(7) Next, the number of reactions and the reaction paths which produce deficit as well as the stochiometric coefficients are read.
(8) The (VKij) matrix for the deficit reactions is formed by accessing the large ($\bar{A}$) on mass storage.
(9) The deficit contribution to each reaction scheme is computed as shown in equation (44) for the oxydation of ammonia.
(10) The resulting deficit and the dissolved oxygen concentration is printed and accumulated for the calculation of total deficit.

SOURCE PROGRAM LISTING

NOTE: The source code and the sample test case can be obtained in a computer readable format for a nominal fee ($20.00 at present) by writing to the office originating this document. This fee is waived for federal agencies or local governmental planning organizations.

```
C     ********************************************************************** MAIN001
C                                                                            MAIN002
C     THIS PROGRAM SOLVES A SYSTEM OF FIRST ORDER CONSECUTIVE REACTION       MAIN003
C     WITH FEEDBACK IN A MULTI-DIMENSIONAL AQUATIC ENVIRONMENT               MAIN004
C                                                                            MAIN005
C              *** FEDBAK03 ***                                              MAIN006
C                                                                            MAIN007
C     THIS VERSION 2.0 CAN SOLVE UP TO A 60 SEGMENT SYSTEM, WHERE THE NO.    MAIN008
C     OF COMPONENTS * NUMBER OF SEGMENTS IS EQUAL OR LESS THAN 180           MAIN009
C                                                                            MAIN010
C     ********************************************************************** MAIN011
      INTEGER OUT                                                            MAIN012
      DIMENSION TITLE(20)                                                    MAIN013
      COMMON/BLK1/NR,N,NP,INPUT,OUT                                          MAIN014
      COMMON/OPTION/IPRINT,ISENS,KWDEF,KBOD,NRUNS                            MAIN015
      DEFINE FILE 1(60,60,U,IREC1)                                           MAIN016
      DEFINE FILE 3(180,180,U,IREC3)                                         MAIN017
C     ********************************************************************** MAIN018
C                                                                           * MAIN019
C     ASSIGN DEVICE NUMBERS TO  CARD READER, LINE PRINTER ...               * MAIN020
C                                                                           * MAIN021
C     ********************************************************************** MAIN022
      INPUT=8                                                                MAIN023
      OUT=6                                                                  MAIN024
      IDODSW=0                                                               MAIN025
      READ(INPUT,200)TITLE                                                   MAIN026
  200 FORMAT (20A4)                                                          MAIN027
C     ********************************************************************** MAIN028
C                                                                           * MAIN029
C     N= NO. OF SEGMENTS (SECTIONS)                                         * MAIN030
C     NR = NO. OF COMPONENTS (REACTANTS)                                    * MAIN031
C                                                                           * MAIN032
C     ********************************************************************** MAIN033
      READ(INPUT,9999)N,NR                                                   MAIN034
9999  FORMAT(2I2)                                                            MAIN035
      WRITE (OUT,201) TITLE,N,NR                                             MAIN036
  201 FORMAT(1H1,20X,20A4//28X,'NUMBER OF SECTIONS = ',I3,10X,'NUMBER OF     MAIN037
     1 REACTION COMPONENTS = ',I2,//)                                        MAIN038
      NP=N*NR                                                                MAIN039
C     ********************************************************************** MAIN040
C                                                                           * MAIN041
C     READ PROGRAM OPTIONS                                                  * MAIN042
C                                                                           * MAIN043
C     ********************************************************************** MAIN044
      READ(INPUT,1101)IPRINT,ISENS,KBOD,KWDEF                                MAIN045
 1101 FORMAT(4I2)                                                            MAIN046
C     ********************************************************************** MAIN047
C                                                                           * MAIN048
C     READ SYSTEM GEOMETRY -- DISPERSIONS                                   * MAIN049
C                                                                           * MAIN050
C     ********************************************************************** MAIN051
      CALL DATA                                                              MAIN052
C     ********************************************************************** MAIN053
C                                                                           * MAIN054
C    CHECK FLOW REGIMES                                                     * MAIN055
C                                                                           * MAIN056
C     ********************************************************************** MAIN057
      CALL FLOW                                                              MAIN058
C     ********************************************************************** MAIN059
```

```
C                                                                       * MAIN060
C                                                                       * MAIN061
C     SET UP SYSTEM MATRIX                                              * MAIN062
C                                                                       * MAIN063
C     ********************************************************************* MAIN064
      CALL  SETUP                                                         MAIN065
C     ********************************************************************* MAIN066
C                                                                       * MAIN067
C     READ AND PRINT LOADS + REACTION RATES -- COMPUTE SOURCE VECTOR    * MAIN068
C                                                                       * MAIN069
C     ********************************************************************* MAIN070
      CALL SOURCE                                                         MAIN071
C     ********************************************************************* MAIN072
C                                                                       * MAIN073
C     NOW FORM LARGE MATRIX OF KV TERMS AND (A) MATRICES                * MAIN074
C                                                                       * MAIN075
C     ********************************************************************* MAIN076
      CALL MATRIX                                                         MAIN077
C     ********************************************************************* MAIN078
C                                                                       * MAIN079
C     COMPUTE INVERSE OF LARGE MATRIX  *  SYSTEM CONCENTRATION VECTOR   * MAIN080
C                                                                       * MAIN081
C     ********************************************************************* MAIN082
      CALL MATN                                                           MAIN083
C     ********************************************************************* MAIN084
C                                                                       * MAIN085
C     TEST FOR DISSOLVED OXYGEN DEFICIT ANALYSIS                        * MAIN086
C                                                                       * MAIN087
C     ********************************************************************* MAIN088
      IF(KBOD.GT.0)CALL DODEF                                             MAIN089
   40 IF(ISENS)100,100,801                                                MAIN090
C     ********************************************************************* MAIN091
C                                                                       * MAIN092
C     READ NUMBER OF SENSITIVITY RUNS DESIRED                           * MAIN093
C     ********************************************************************* MAIN094
  801 READ(INPUT,301)NRUNS                                                MAIN095
  301 FORMAT(I2)                                                          MAIN096
  302 CALL SENS                                                           MAIN097
  304 CALL MATN                                                           MAIN098
  310 IF(KBOD.GT.0)CALL DODEF                                             MAIN099
      IF(NRUNS.GT.0)GO TO 302                                             MAIN100
  100 CALL EXIT                                                           MAIN101
      END                                                                 DATA001
      SUBROUTINE DATA                                                     DATA002
C*********************************************************************** DATA003
C                                                                       * DATA004
C                          SUBROUTINE DATA                              * DATA005
C                                                                       * DATA006
C     INPUTS DATA WHICH WOULD NOT VARY WITH THE PARTICULAR              * DATA007
C     CONSTITUENT BEING MODELED, I.E. PRIMARILY PHYSICAL AND            * DATA008
C     HYDRODYNAMIC DATA                                                 * DATA009
C                                                                       * DATA010
C*********************************************************************** DATA011
C*********************************************************************** DATA012
      INTEGER OUT                                                         DATA013
      REAL LA(60,6)                                                       DATA014
      DIMENSION AREA(6),E(6),Q(6)                                         DATA015
      COMMON/BLK1/NR,N,NP,INPUT,OUT                                       DATA016
      COMMON/BLK3/VOL(60),TEMPER(60),CHLOR(60)                            DATA017
```

```
      COMMON/COHOE/LA,SCALE(4),ARRAY(60,18)                                 DATA018
      COMMON/OPTION/IPRINT,ISENS,KWDEF,KBOD                                 DATA019
      COMMON/UT1/IARAY(60,6)                                                DATA020
      COMMON/BLK4/NLA(60,6)                                                 DATA021
   31 FORMAT(8F10.0)                                                        DATA022
  100 FORMAT(4F7.0)                                                         DATA023
  101 FORMAT (///28X, 'ZERO SEGMENT NUMBER IN INTERFACE ', I3, '-', I3,     DATA024
     1' COMPUTATION DISCONTINUED')                                          DATA025
  200 FORMAT(6(I4,' -',I4,2X,F7.0,2X))                                      DATA026
  201 FORMAT(6(I3,F10.0))                                                   DATA027
  300 FORMAT(                 '1                                            DATA028
     1 CHARAC. LENGTHS OF SEGMENTS (FT) '///' INTERFACE  LENGTH    INTER   DATA029
     1FACE  LENGTH    INTERFACE  LENGTH    INTERFACE  LENGTH    INTERFAC   DATA030
     2E  LENGTH    INTERFACE  LENGTH')                                      DATA031
 1111 FORMAT(3(3F6.0,I3))                                                   DATA032
 2002 FORMAT(//10X,'**SCALE FACTORS**'//10X,                                DATA033
     1'SCALE(1) = ',F10.3,4X,'SCALE(2) = ',F10.3,4X,                        DATA034
     2'SCALE(3) = ',F10.3,4X,'SCALE(4) = ',F10.3 )                          DATA035
 2200 FORMAT('1',2X,'INTERFACE',5X,'AREA',7X,'E',8X,'Q ',2(6X,'INTERFACE    DATA036
     1',5X,'AREA',9X,'E',7X,'Q')/,4X,'ROW-SEG',4X,'(FT**2)', 2X,            DATA037
     2'(MI**2/D)',2X,'(CFS)',   6X,'ROW-SEG',5X,'(FT**2)',2X,               DATA038
     3'(MI**2/D)',2X,'(CFS)',5X,'ROW-SEG',5X,'(FT**2)',2X,'(MI**2/D)',      DATA039
     4 2X,'(CFS)' )                                                         DATA040
 3000 FORMAT(  '   (',I3,'-',I3,')',2X,F8.0,4X,F6.3,F8.1,2(5X,'(',I3,       DATA041
     1 '-',I3,')',2X,F8.0,4X,F6.3,F8.1))                                    DATA042
C*************************************************************************** DATA043
C                                                                           * DATA044
C     READ AND WRITE THE SYSTEM PARAMETERS AND SCALE FACTORS                * DATA045
C                                                                           * DATA046
C*************************************************************************** DATA047
      READ(INPUT,100) SCALE                                                 DATA048
      WRITE(OUT,2002)SCALE                                                  DATA049
C*************************************************************************** DATA050
C                                                                           * DATA051
C     INITIALIZE MATRICES TO ZERO                                           * DATA052
C                                                                           * DATA053
C*************************************************************************** DATA054
      DO 1001 I=1,N                                                         DATA055
      DO 1021 J=1,18                                                        DATA056
 1021 ARRAY(I,J)=0.0                                                        DATA057
      DO 1001 J=1,6                                                         DATA058
      LA(I,J)=0.0                                                           DATA059
 1001 IARAY(I,J)=0                                                          DATA060
C*************************************************************************** DATA061
C                                                                           * DATA062
C     READ AND WRITE INTERFACE PARAMETERS AREA, E AND Q AND PLACE           * DATA063
C     THEM INTO ARRAY                                                       * DATA064
C                                                                           * DATA065
C*************************************************************************** DATA066
      DO 1002 I=1,N                                                         DATA067
      READ(INPUT,1111) (AREA(J),E(J),Q(J),     IARAY(I,J),J=1,6)            DATA068
      DO 1002 J=1,6                                                         DATA069
      IF(AREA(J))72,71,72                                                   DATA070
   72 IF(IARAY(I,J))74,74,71                                                DATA071
   74 WRITE (OUT,101)I,IARAY(I,J)                                           DATA072
      CALL EXIT                                                             DATA073
   71 JJ=(J-1)*3+1                                                          DATA074
      ARRAY(I,JJ)=AREA(J)*SCALE(1)                                          DATA075
      ARRAY(I,JJ+1)=E(J)*SCALE(2)                                           DATA076
```

```
 1002 ARRAY(I,JJ+2)=C(J)*SCALE(3)                                          DATA077
C     ************************************************************      * DATA078
C                                                                       * DATA079
C     PROPAGATE AREAS,DISPERSIONS AND FLOWS FOR INTERFACES AT (J,I)     * DATA080
C     USING INPUT VALUES OF THE (I,J) INTERFACE                         * DATA081
C                                                                       * DATA082
C*************************************************************************DATA083
      DO 1008 I=1,N                                                       DATA084
      DO 1008 JJ=1,6                                                      DATA085
      IF(IARAY(I,JJ))1008,1008,1088                                       DATA086
 1088 IF(IARAY(I,JJ)-I)1012,1008,1012                                     DATA087
 1012 JJJ=(JJ-1)*3+1                                                      DATA088
      DO 1004 J=1,6                                                       DATA089
      JK=(J-1)*3+1                                                        DATA090
      NK=IARAY(I,JJ)                                                      DATA091
      IF(IARAY(NK,          J))1005,1006,1005                             DATA092
 1006 ARRAY(NK,          JK)=ARRAY(I,JJJ)                                 DATA093
      ARRAY(NK,          JK+1)=ARRAY(I,JJJ+1)                             DATA094
      ARRAY(NK,          JK+2)=-ARRAY(I,JJJ+2)                            DATA095
      IARAY(NK,          J)=I                                             DATA096
      GO TO 1008                                                          DATA097
 1005 IF(IARAY(NK,          J)-I          )1004,1008,1004                 DATA098
 1004 CONTINUE                                                            DATA099
 1008 CONTINUE                                                            DATA100
      WRITE (OUT,2200)                                                    DATA101
      LINE=0                                                              DATA102
      DO 1032 I=1,N                                                       DATA103
      WRITE(OUT,3000)  I,IARAY(I,1),ARRAY(I,1),ARRAY(I,2),ARRAY(I,3),     DATA104
     1                I,IARAY(I,2),ARRAY(I,4),ARRAY(I,5),ARRAY(I,6),     DATA105
     1                I,IARAY(I,3),ARRAY(I,7),ARRAY(I,8),ARRAY(I,9)      DATA106
      WRITE(OUT,3000)  I,IARAY(I,4),ARRAY(I,10),ARRAY(I,11),ARRAY(I,12), DATA107
     1                I,IARAY(I,5),ARRAY(I,13),ARRAY(I,14),ARRAY(I,15),  DATA108
     1                I,IARAY(I,6),ARRAY(I,16),ARRAY(I,17),ARRAY(I,18)   DATA109
      LINE=LINE+2                                                         DATA110
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)1032,1099,1099                    DATA111
 1099 WRITE(OUT,2200)                                                     DATA112
      LINE=0                                                              DATA113
 1032 CONTINUE                                                            DATA114
C*************************************************************************DATA115
C                                                                       * DATA116
C     READ AND WRITE INTERFACE PARAMETER- CHARACTERISTIC LENGTH         * DATA117
C                                                                       * DATA118
C*************************************************************************DATA119
      WRITE(OUT,300)                                                      DATA120
      LINE=0                                                              DATA121
      DO 500 I=1,N                                                        DATA122
      READ(INPUT,201)(NLA(I,J),LA(I,J), J=1,6)                            DATA123
      DO 77 J=1,6                                                         DATA124
   77 LA(I,J)=LA(I,J)*SCALE(4)                                            DATA125
      WRITE(OUT,200)(I,NLA(I,J),LA(I,J),J=1,6)                            DATA126
      LINE=LINE+1                                                         DATA127
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)500,1098,1098                     DATA128
 1098 WRITE(OUT,300)                                                      DATA129
      LINE=0                                                              DATA130
  500 CONTINUE                                                            DATA131
C*************************************************************************DATA132
C                                                                       * DATA133
C     READ CHLORIDE CONCENTRATION, TEMPERATURE AND VOLUME AT EA. SECT.  * DATA134
C                                                                       * DATA135
```

```
C*************************************************************************** DATA136
      IF(KBOD.GT.0)READ(INPUT,31)(CHLOR(K),K=1,N)                           DATA137
      READ(INPUT,31)  (TEMPER(I),VOL(I),I=1,N)                              DATA138
      IF(TEMPER(2))14,13,14                                                 DATA139
   13 DO 14 I=1,N                                                           DATA140
      TEMPER(I)=TEMPER(1)                                                   DATA141
   14 CONTINUE                                                              DATA142
C*************************************************************************** DATA143
C                                                                           * DATA144
C     APPLY CONVERSION FACTORS TO FLOW AND VOLUME                           * DATA145
C                                                                           * DATA146
C*************************************************************************** DATA147
      DO 12 I=1,N                                                           DATA148
      DO 23 J=1,6                                                           DATA149
      JJ=(J-1)*3+1                                                          DATA150
   23 ARRAY(I,JJ+2)=ARRAY(I,JJ+2)*.6463                                     DATA151
   12 VOL(I)=VOL(I)*7.48                                                    DATA152
      RETURN                                                                DATA153
      END                                                                   DATA154
      SUBROUTINE FLOW                                                       FLOW001
C*************************************************************************** FLOW002
C*************************************************************************** FLOW003
C                                                                           * FLOW004
C                          SUBROUTINE FLOW                                  * FLOW005
C                                                                           * FLOW006
C     THIS SUBROUTINE CALCULATES A FLOW BALANCE AROUND EACH SECTION         * FLOW007
C     TO INSURE THAT THE FLOW REGIME HAS BEEN INPUT CORRECTLY               * FLOW008
C                                                                           * FLOW009
C*************************************************************************** FLOW010
C*************************************************************************** FLOW011
      REAL LA(60,6)                                                         FLOW012
      INTEGER OUT                                                           FLOW013
      COMMON/BLK1/NR,N,NP,INPUT,OUT                                         FLOW014
      COMMON/COHOE/LA,SCALE(4),ARRAY(60,18)                                 FLOW015
      NMN=0                                                                 FLOW016
  214 FORMAT('       SECTION ',I3,' HAS AN EXCESS FLOW   OF ',E12.6,        FLOW017
     1' CFS')                                                               FLOW018
  302 FORMAT('1')                                                           FLOW019
      DO 213 I=1,N                                                          FLOW020
      QQ=ARRAY(I,3)+ARRAY(I,6)+ARRAY(I,9)+ARRAY(I,12)+ARRAY(I,15)           FLOW021
     1+ARRAY(I,18)                                                          FLOW022
      QQ=QQ/.6463                                                           FLOW023
      IF(QQ-.001)216,37,37                                                  FLOW024
  216 IF(QQ+.001)37,37,213                                                  FLOW025
   37 IF(NMN)29,29,27                                                       FLOW026
   29 NMN=1                                                                 FLOW027
      WRITE(OUT,302)                                                        FLOW028
   27 WRITE(OUT,214)I,QQ                                                    FLOW029
  213 CONTINUE                                                              FLOW030
      RETURN                                                                FLOW031
      END                                                                   FLOW032
      SUBROUTINE SETUP                                                      SETUP001
C*************************************************************************** SETUP002
C*************************************************************************** SETUP003
C                                                                           *SETUP004
C                          SUBROUTINE SETUP                                 *SETUP005
C                                                                           *SETUP006
C     THE PRIMARY TASK OF SETUP IS TO CONSTRUCT THE FLOWNET MATRIX (A)      *SETUP007
C                                                                           *SETUP008
C
```

```
C*******************************************************************************SETUP009
C*******************************************************************************SETUP010
      REAL LA(60,6)                                                              SETUP011
      INTEGER OUT                                                                SETUP012
      DIMENSION EPRIM(6),ALPHA(6)                                                SETUP013
      COMMON/BLK1/NR,N,NP,INPUT,OUT,AB(60,60)                                    SETUP014
      COMMON/BLK3/VOL(60),T(60),CHLOR(60)                                        SETUP015
      COMMON/COMCE/LA,SCALE(4),ARRAY(60,18)                                      SETUP016
      COMMON/OPTION/IPRINT,ISENS,KWDEF,KBOD                                      SETUP017
      COMMON/UT1/IARAY(60,6)                                                     SETUP018
      COMMON/BLK4/NLA(60,6)                                                      SETUP019
   23 FORMAT(6(I3,'-',I3,F6.0,F6.3,2X))                                          SETUP020
   24 FORMAT('1',49X,'VALUES OF ALPHA AND EPRIM',///,6(' INTRFC EPRIM ALSETUP021
     1PHA  '),/,6(8X,'(MGD)',8X) )                                               SETUP022
   42 FORMAT(4X,I4,9X,F12.2,16X,F5.2,23X,F10.0)                                  SETUP023
   43 FORMAT('1',3X,'SECTION ',8X,'VOLUME (MGAL)',8X,                           SETUP024
     1'TEMPERATURE (CENT.)',8X,'CHLORIDE CONC (MG/L)'  )                         SETUP025
   44 FORMAT('1',3X,'SECTION ',8X,'VOLUME (MGAL)',8X,                           SETUP026
     1'TEMPERATURE (CENT.)')                                                     SETUP027
  100 FORMAT(///,10X,'SEGMENT (',I3,'-',I3,') IS A BOUNDARY - BUT NO LENSETUP028
     1GTH WAS SPECIFIED')                                                        SETUP029
  101 FORMAT(//,10X,'PROGRAM TERMINATING DUE TO ',I3,' BOUNDARY CONDITIOSETUP030
     1NS WITH MISSING LENGTHS')                                                  SETUP031
      WRITE (OUT,24)                                                           SETUP032
      IEXIT=0                                                                  SETUP033
      DO 28 I=1,N                                                              SETUP034
      DO 99 J=1,N                                                              SETUP035
   99 AB(I,J)=0.0                                                              SETUP036
C*******************************************************************************SETUP037
C                                                                              *SETUP038
C     CALCULATE EPRIM AND ALPHA FOR EACH INTERFACE                             *SETUP039
C                                                                              *SETUP040
C*******************************************************************************SETUP041
      DO 20 J=1,6                                                              SETUP042
      EPRIM(J)=0.0                                                             SETUP043
      ALPHA(J)=0.0                                                             SETUP044
      JJ=(J-1)*3+1                                                             SETUP045
      K=IARAY(I,J)                                                             SETUP046
      IF(K)20,20,10                                                            SETUP047
C*******************************************************************************SETUP048
C                                                                              *SETUP049
C     GENERATE (I,K) LENGTH ELEMENT                                            *SETUP050
C                                                                              *SETUP051
C*******************************************************************************SETUP052
   10 XLEN1=0.0                                                                SETUP053
      DO 77  KK=1,6                                                            SETUP054
      IF(NLA(I,KK)-K)77,81,77                                                  SETUP055
   81 XLEN1=LA(I,KK)                                                           SETUP056
      GO TO 82                                                                 SETUP057
   77 CONTINUE                                                                 SETUP058
C*******************************************************************************SETUP059
C                                                                              *SETUP060
C     GENERATE (K,I) LENGTH ELEMENT                                            *SETUP061
C                                                                              *SETUP062
C*******************************************************************************SETUP063
   82 XLEN2=0.0                                                                SETUP064
      DO 78 KK=1,6                                                             SETUP065
      IF(NLA(K,KK)-I)78,84,78                                                  SETUP066
   84 XLEN2=LA(K,KK)                                                           SETUP067
```

```
      GO TO 83                                                          SETUP068
   78 CONTINUE                                                          SETUP069
   83 IF(XLEN1+XLEN2)87,87,85                                           SETUP070
   87 WRITE(OUT,100)I,K                                                 SETUP071
      IEXIT=IEXIT+1                                                     SETUP072
      GO TO 17                                                          SETUP073
   85 EPRIM(J)=ARRAY(I,JJ)*ARRAY(I,JJ+1)*417.1166/(XLEN2+XLEN1)         SETUP074
      IF(ARRAY(I,JJ+2))300,301,301                                      SETUP075
  300 ALPHA(J)=XLEN1/(XLEN1+XLEN2)                                      SETUP076
      GO TO 200                                                         SETUP077
  301 ALPHA(J)=XLEN2/(XLEN1+XLEN2)                                      SETUP078
  200 CONTINUE                                                          SETUP079
      IF(ARRAY(I,JJ+2))14,17,14                                         SETUP080
   14 IF(ALPHA(J)-1.0+EPRIM(J)/ABS(ARRAY(I,JJ+2)))15,15,17              SETUP081
   15 ALPHA(J)=1.0-(EPRIM(J)/(2.*ABS(ARRAY(I,JJ+2))))                   SETUP082
C*************************************************************************SETUP083
C                                                                       *SETUP084
C     SETUP THE SYSTEM FLOWNET MATRIX (A)                               *SETUP085
C                                                                       *SETUP086
C*************************************************************************SETUP087
   17 IF(ARRAY(I,JJ+2))18,19,19                                         SETUP088
   18 AB(I,I)=AB(I,I)+EPRIM(J)+(1.0-ALPHA(J))*ARRAY(I,JJ+2)             SETUP089
      IF(I-K)21,20,21                                                   SETUP090
   21 AB(I,K)=ALPHA(J)*ARRAY(I,JJ+2) - EPRIM(J)                         SETUP091
      GO TO 20                                                          SETUP092
   19 AB(I,I)=AB(I,I)+ALPHA(J)*ARRAY(I,JJ+2)+ EPRIM(J)                  SETUP093
      IF(I-K) 22,20,22                                                  SETUP094
   22 AB(I,K)=(1.-ALPHA(J))*ARRAY(I,JJ+2)-EPRIM(J)                      SETUP095
   20 CONTINUE                                                          SETUP096
C*************************************************************************SETUP097
C                                                                       *SETUP098
C     WRITE VALUES OF EPRIM AND ALPHA                                   *SETUP099
C                                                                       *SETUP100
C*************************************************************************SETUP101
      WRITE(OUT,23)                                                     SETUP102
     1I,IARAY(I,1),EPRIM(1),ALPHA(1),I,IARAY(I,2),EPRIM(2),ALPHA(2),    SETUP103
     2I,IARAY(I,3),EPRIM(3),ALPHA(3),I,IARAY(I,4),EPRIM(4),ALPHA(4),    SETUP104
     2I,IARAY(I,5),EPRIM(5),ALPHA(5),I,IARAY(I,6),EPRIM(6),ALPHA(6)     SETUP105
      LINE=LINE+1                                                       SETUP106
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)28,1028,1028                    SETUP107
 1028 WRITE(OUT,23)                                                     SETUP108
      LINE=0                                                            SETUP109
   28 CONTINUE                                                          SETUP110
C*************************************************************************SETUP111
C                                                                       *SETUP112
C     WRITE FLOWNET MATRIX ON MASS STORAGE DISK FILE 1                  *SETUP113
C                                                                       *SETUP114
C*************************************************************************SETUP115
      DO 30  I=1,N                                                      SETUP116
      WRITE(1'I)(AB(I,J),J=1,N)                                         SETUP117
   30 CONTINUE                                                          SETUP118
C*************************************************************************SETUP119
C                                                                       *SETUP120
C     WRITE SECTION PARAMETERS                                          *SETUP121
C                                                                       *SETUP122
C*************************************************************************SETUP123
      IF(KBOD.GT.0)WRITE(OUT,43)                                        SETUP124
      IF(KBOD.LE.0)WRITE(OUT,44)                                        SETUP125
      LINE=0                                                            SETUP126
```

```
      DO 399 I=1,N                                                      SETUP127
      IF(KBOD.GT.0) WRITE(OUT,42) I,VOL(I),T(I),CHLOR(I)                SETUP128
      IF(KBOD.LE.0) WRITE(OUT,42) I,VOL(I),T(I)                         SETUP129
      LINE=LINE+1                                                       SETUP130
      IF(FLOAT(LINE/54)-FLOAT(LINE)/54.)399,1029,1029                   SETUP131
 1029 IF(KBOD.GT.0)WRITE(OUT,43)                                        SETUP132
      IF(KBOD.LE.0)WRITE(OUT,44)                                        SETUP133
      LINE=0                                                            SETUP134
  399 CONTINUE                                                          SETUP135
      IF(IEXIT)88,88,89                                                 SETUP136
   89 WRITE(OUT,101)IEXIT                                               SETUP137
      CALL EXIT                                                         SETUP138
   88 RETURN                                                            SETUP139
      END                                                               SETUP140
      SUBROUTINE SOURCE                                                 SOURC001
C     ************************************************************** *SOURC002
C     **************************************************************SOURC003
C                                                                      *SOURC004
C     THIS SUBROUTINE WILL READ REACTION SCHEMES AND THE CORRESPONDING *SOURC005
C     REACTION RATE CONSTANTS FOR EACH SYSTEM SEGMENT, AND CONSTRUCT   *SOURC006
C     THE (VK) SYSTEM MATRIX.                                          *SOURC007
C                                                                      *SOURC008
C     IT WILL ALSO READ ALL DIRECT DISCHARGES AND BOUNDARY CONDITIONS  *SOURC009
C     AND FORM THE SYSTEM SOURCE VECTOR (W)                            *SOURC010
C     **************************************************************SOURC011
C                                                                      *SOURC012
      REAL LOAD,K,LA(60,6)                                              SOURC013
      INTEGER OUT                                                       SOURC014
      DIMENSION X(60),EPRIM(6),ALPHA(6)                                 SOURC015
      COMMON/UT1/IARAY(60,6)                                            SOURC016
      COMMON/BLK4/NLA(60,6)                                             SOURC017
      COMMON/COHOE/LA,SCALE(4),ARRAY(60,18)                             SOURC018
      COMMON/UT2/B(180),K(60)                                           SOURC019
      COMMON/BLK1/N,NS,NP,INPUT,OUT,DUMMY(60,60)                        SOURC020
      COMMON/BLK5/W(180)                                                SOURC021
      COMMON/BLK3/V(60),TEMPER(60),CHLOR(60)                            SOURC022
      EQUIVALENCE(X(1),DUMMY(1,4))                                      SOURC023
      WRITE(OUT,80)(II,II=1,NS)                                         SOURC024
   80 FORMAT('1',50X,'SYSTEM REACTION RATE CONSTANTS (1/DAY)',//,25X,   SOURC025
     1'S  E  C  T  I  O  N    N  U  M  B  E  R  ',/,' REACTION',3X,      SOURC026
     2 'THETA',5X,13I8,/,4(21X,13I8,/))                                 SOURC027
      DO 53 II=1,NP                                                     SOURC028
      DO 52 JJ=1,NP                                                     SOURC029
   52 B(JJ)=0.0                                                         SOURC030
   53 WRITE(3'II)(B(JK),JK=1,NP)                                        SOURC031
C     REACTION COEFFICIENTS (K'S) WILL BE INPUT.                       *SOURC032
C     BLANK CARD STOPS K INPUT.                                        *SOURC033
  200 READ(INPUT,2)I,J,THETA                                            SOURC034
    2 FORMAT(2I2,1X,F5.3)                                               SOURC035
      IF (I)89,89,90                                                    SOURC036
   90 READ(INPUT,4)(K(L),L=1,NS)                                        SOURC037
C     TEMPERATURE CORRECT REACTION RATE CONSTANTS                      *SOURC038
      IF(THETA.LE.0)GO TO 82                                            SOURC039
      DO 81  L=1,NS                                                     SOURC040
   81 K(L)=K(L)*THETA**(TEMPER(L)-20.)                                  SOURC041
    4 FORMAT (8F10.0)                                                   SOURC042
   82 WRITE(OUT,5)I,J,THETA,(K(L),L=1,NS)                               SOURC043
    5 FORMAT(1X,I2,' TO ',I2,3X,F5.3,5X,13F8.4,/,4(22X,13F8.4,/))       SOURC044
C     **************************************************************SOURC045
```

```
C                                                                       *SOURC046
C     WRITE (VK) COEFFICIENTS FOR CORRESPONDING REACTION ON MASS        *SOURC047
C     STORAGE FILE 3                                                    *SOURC048
C                                                                       *SOURC049
C     ************************************************************************SOURC050
      DO 100 L=1,NS                                                      SOURC051
      IP=NS*(J-1)+L                                                      SOURC052
      READ(3'IP)(B(II),II=1,NP)                                          SOURC053
      JP=NS*(I-1)+L                                                      SOURC054
      B(JP)=V(L)*K(L)                                                    SOURC055
      IF(I-J)101,95,101                                                  SOURC056
  101 B(JP)=-B(JP)                                                       SOURC057
   95 WRITE(3'IP)(B(IC),IC=1,NP)                                         SOURC058
  100 CONTINUE                                                           SOURC059
      GO TO 200                                                          SOURC060
C************************************************************************SOURC061
C                                                                       *SOURC062
C     READ  DIRECT COMPONENT DISCHARGES INTO THE SYSTEM                 *SOURC063
C                                                                       *SOURC064
C************************************************************************SOURC065
   89 DO 61 JK=1,NP                                                      SOURC066
   61 W(JK)=0.0                                                          SOURC067
      WRITE(OUT,36)                                                      SOURC068
   36 FORMAT('1',/,53X,'DIRECT DISCHARGES (LB/DAY)',                     SOURC069
     1          //,53X,'LOAD',5X,'SECTION',4X,'COMPONENT')              SOURC070
C    LOADS WILL BE INPUT                                               *SOURC071
C    BLANK CARD STOPS LOAD INPUT.                                      *SOURC072
  201 READ(INPUT,25)LOAD,ISEC,IREAC                                      SOURC073
   25 FORMAT(F10.0,2I2)                                                  SOURC074
      IF(IREAC)99,99,26                                                  SOURC075
C************************************************************************SOURC076
C                                                                       *SOURC077
C    FORMATION OF SOURCE VECTOR.                                        *SOURC078
C                                                                       *SOURC079
C************************************************************************SOURC080
   26 IPOS=NS*(IREAC-1)+ISEC                                             SOURC081
      W(IPOS)=LOAD                                                       SOURC082
      WRITE(OUT,33)LOAD,ISEC,IREAC                                       SOURC083
   33 FORMAT(50X,F10.1,6X,I2,8X,I2)                                      SOURC084
      GO TO 201                                                          SOURC085
   99 I1=1                                                               SOURC086
      WRITE(OUT,76)                                                      SOURC087
   76 FORMAT('1',50X,'SYSTEM BOUNDARY CONCENTRATIONS',//,40X,'SECTION',8SOURC088
     1X,'COMPONENT',8X,'BOUND. CONC. (MG/L)')                            SOURC089
   75 READ(INPUT,71)BC,ISYS,ISECT                                        SOURC090
   71 FORMAT(F10.0,2I2)                                                  SOURC091
      IF(ISYS)999,999,73                                                 SOURC092
   73 IWP=NS*(ISYS-1)+ISECT                                              SOURC093
C************************************************************************SOURC094
C                                                                       *SOURC095
C     CALCULATE THE PART OF THE FORCING FUNCTION DUE TO BOUNDARY COND.  *SOURC096
C                                                                       *SOURC097
C************************************************************************SOURC098
      I=ISECT                                                            SOURC099
      DO 30 J=1,6                                                        SOURC100
      JJ=(J-1)*3+1                                                       SOURC101
      L=IARAY(I,J)                                                       SOURC102
      IF(L-I) 30,35,30                                                   SOURC103
   30 CONTINUE                                                           SOURC104
```

```
      GO TO 3                                                                SOURC105
C*****************************************************************************SOURC106
C                                                                            *SOURC107
C     GENERATE LA(K,K) LENGTH ELEMENT                                        *SOURC108
C                                                                            *SOURC109
C*****************************************************************************SOURC110
   35 XLEN1=0.0                                                              SOURC111
      DO 77  KK=1,6                                                          SOURC112
      IF(NLA(I,KK)-L)77,88,77                                                SOURC113
   77 CONTINUE                                                               SOURC114
    3 WRITE(OUT,102)                                                         SOURC115
      CALL EXIT                                                              SOURC116
  102 FORMAT(//,10X,'PROGRAM TERMINATING DUE TO  A     BOUNDARY CONDITIOSOURC117
     1NS WITH MISSING LENGTHS')                                              SOURC118
   88 XLEN1=LA(I,KK)                                                         SOURC119
      XLEN2=LA(I,KK)                                                         SOURC120
   83 ALPHA(J)=0.5                                                           SOURC121
      EPRIM(J)=ARRAY(I,JJ)*ARRAY(I,JJ+1)*417.1166/(XLEN2+XLEN1)              SOURC122
      IF(-0.5+EPRIM(J)/ABS(ARRAY(I,JJ+2)))40,45,45                           SOURC123
   40 ALPHA(J)=1.0-EPRIM(J)/(2.0*ABS(ARRAY(I,JJ+2)))                         SOURC124
   45 IF(ARRAY(I,JJ+2))50,55,55                                              SOURC125
   50 W(IWP)=W(IWP)+BC*8.34*(EPRIM(J)-ALPHA(J)*ARRAY(I,JJ+2))                SOURC126
      KLAST=KLAST+1                                                          SOURC127
      GO TO 60                                                               SOURC128
   55 W(IWP)=W(IWP)+BC*8.34*(EPRIM(J)-(1.0-ALPHA(J))*ARRAY(I,JJ+2))          SOURC129
      KLAST=KLAST+1                                                          SOURC130
   60 WRITE(OUT,74)ISECT,ISYS,BC                                             SOURC131
   74 FORMAT(40X,I4,13X,I3,14X,F8.3)                                         SOURC132
      GO TO 75                                                               SOURC133
  999 WRITE(OUT,85)                                                          SOURC134
   85 FORMAT('1',50X,'SYSTEM SOURCE VECTOR',//,40X,'COMPONENT',5X,           SOURC135
     1'SECTION',5X,'LOAD (LB/DAY)')                                          SOURC136
      II=1                                                                   SOURC137
      DO 98 I=1,N                                                            SOURC138
      DO 98 J=1,NS                                                           SOURC139
      WRITE(OUT,86)I,J,W(II)                                                 SOURC140
   86 FORMAT(42X,I3, 9X,I3, 9X,F12.1)                                        SOURC141
      II = II + 1                                                            SOURC142
   98 CONTINUE                                                               SOURC143
      RETURN                                                                 SOURC144
      END                                                                    SOURC145
      SUBROUTINE MATRIX                                                      MATRX001
C     *****************************************************************************MATRX002
C                                                                           *MATRX003
C     THIS ROUTINE WILL COMBINE THE VK MATRICES ON MASS STORAGE WITH        *MATRX004
C     THE FLOWNET (A) MATRICES TO FORM THE MATRIX OF MATRICES (A)           *MATRX005
C                                                                           *MATRX006
C     *****************************************************************************MATRX007
      INTEGER OUT                                                            MATRX008
      COMMON/UT1/A(60),B(180)                                                MATRX009
      COMMON/BLK1/N,NS,NP,INPUT,OUT,AC(180,180)                              MATRX010
      COMMON/BLK4/PTMIL(60)                                                  MATRX011
      COMMON/BLK5/WS(180)                                                    MATRX012
      COMMON/OPTION/IPRINT                                                   MATRX013
      DO 115 I=1,NS                                                          MATRX014
      READ(1'I)A                                                             MATRX015
      DO 20 J=1,N                                                            MATRX016
      JJ=NS*(J-1)+ I                                                         MATRX017
      READ(3'JJ)B                                                            MATRX018
```

```
      DO 60 L=1,NS                                                      MATRX019
      LL = NS *(J-1)+L                                                  MATRX020
C*****************************************************************************MATRX021
C                                                                      *MATRX022
C    FORMATION OF THE LARGE MATRIX WHICH CONSISTS OF KV AND A MATRICES.*MATRX023
C                                                                      *MATRX024
C*****************************************************************************MATRX025
   60 B(LL)=B(LL)+A(L)                                                  MATRX026
      DO 93 JC=1,NP                                                    MATRX027
   93 AC(JJ,JC)=B(JC)                                                  MATRX028
   20 CONTINUE                                                         MATRX029
  115 CONTINUE                                                         MATRX030
      IF(IPRINT)90,90,70                                               MATRX031
   70 WRITE(OUT,80)                                                    MATRX032
   80 FORMAT(1H1,///,50X,'SYSTEM MATRIX BEFORE SOLUTION',///)          MATRX033
      WRITE(OUT,89)(I,I=1,NP)                                          MATRX034
   89 FORMAT(/,12X,10I12,/)                                            MATRX035
      DO 39 IR=1,NP                                                    MATRX036
      WRITE(OUT,12)IR,(AC(IR,JC),JC=1,NP)                              MATRX037
   12 FORMAT(1X,'ROW NO.',I3,1X,10F12.2/(12X,10F12.2))                 MATRX038
   39 CONTINUE                                                         MATRX039
C*****************************************************************************MATRX040
C                                                                      *MATRX041
C     PTMIL IS AN ARRAY CONSISTING OF THE MILEAGE POINTS OF THE SECTION*MATRX042
C                                                                      *MATRX043
C*****************************************************************************MATRX044
   90 READ (INPUT, 200)(PTMIL(I),I=1,NS)                               MATRX045
  200 FORMAT (8F10.0)                                                  MATRX046
      RETURN                                                           MATRX047
      END                                                              MATRX048
      SUBROUTINE MATN                                                   MATN001
C*****************************************************************************  MATN002
C                                                                     *  MATN003
C                            SUBROUTINE MATN                          *  MATN004
C                                                                     *  MATN005
C     THIS SUBROUTINE INVERTS THE MATRIX AC AND MULTIPLIES            *  MATN006
C     IT BY THE FORCING FUNCTION WS(I)  GIVING SOLUTION VECTOR B(I)   *  MATN007
C                                                                     *  MATN008
C*****************************************************************************  MATN009
C*****************************************************************************  MATN010
      INTEGER OUT                                                       MATN011
      COMMON/UT1/IPIVO(180),PIVOT(180)                                  MATN012
      COMMON/UT2/INDEX(180,2)                                           MATN013
      COMMON/BLK1/N1,N2,N,INPUT,OUT,AC(180,180)                         MATN014
      COMMON/OPTION/IPRINT                                              MATN015
      COMMON/BLK4/PTMIL(60),B(180)                                      MATN016
      COMMON/BLK5/WS(180)                                               MATN017
      DO 900 I=1,N                                                      MATN018
      DO 900 J=1,N                                                      MATN019
  900 AC(I,J)=AC(I,J)/1000000.                                          MATN020
      DO 20 J=1,N                                                       MATN021
      IPIVO(J)=0                                                        MATN022
      DO 21 I=1,2                                                       MATN023
   21 INDEX(J,I)=0                                                      MATN024
   20 CONTINUE                                                          MATN025
      DO 550 I=1,N                                                      MATN026
C                                                                     *  MATN027
C     SEARCH FOR PIVOT ELEMENT                                        *  MATN028
C                                                                     *  MATN029
```

```
      AMAX=0.0                                                          MATN030
      DO 105 J=1,N                                                      MATN031
      IF (IPIVO (J)-1) 60, 105, 60                                      MATN032
   60 DO 100 M=1,N                                                      MATN033
      IF (IPIVO (M)-1) 80, 100, 740                                     MATN034
   80 IF (ABS(AMAX)-ABS(AC(J,M)))85, 100, 100                           MATN035
   85 IROW=J                                                            MATN036
      ICOLU =M                                                          MATN037
      AMAX=AC(J,M)                                                      MATN038
  100 CONTINUE                                                          MATN039
  105 CONTINUE                                                          MATN040
      IPIVO (ICOLU )=IPIVO (ICOLU )+1                                   MATN041
C                                                                     * MATN042
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL               * MATN043
C                                                                     * MATN044
      IF (IROW-ICOLU ) 150, 260, 150                                    MATN045
  150 DO 200 L=1,N                                                      MATN046
      SWAP=AC(IROW,L)                                                   MATN047
      AC(IROW,L)=AC(ICOLU,L)                                            MATN048
  200 AC(ICOLU,L)=SWAP                                                  MATN049
  260 INDEX(I,1)=IROW                                                   MATN050
      INDEX(I,2)=ICOLU                                                  MATN051
      PIVOT(I)=AC(ICOLU,ICOLU)                                          MATN052
C                                                                     * MATN053
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                               * MATN054
C                                                                     * MATN055
      AC(ICOLU,ICOLU)=1.0                                               MATN056
      DO 350 L=1,N                                                      MATN057
  350 AC(ICOLU,L)=AC(ICOLU,L)/PIVOT(I)                                  MATN058
C                                                                     * MATN059
C     REDUCE NON-PIVOT ROWS                                           * MATN060
C                                                                     * MATN061
      DO 550 L1=1,N                                                     MATN062
      IF(L1-ICOLU ) 400, 550, 400                                       MATN063
  400 S=AC(L1,ICOLU)                                                    MATN064
      AC(L1,ICOLU)=0.0                                                  MATN065
      DO 450 L=1,N                                                      MATN066
  450 AC(L1,L)=AC(L1,L)-AC(ICOLU,L)*S                                   MATN067
  550 CONTINUE                                                          MATN068
C                                                                     * MATN069
C     INTERCHANGE COLUMNS                                             * MATN070
C                                                                     * MATN071
      DO 710 I=1,N                                                      MATN072
      L=N+1-I                                                           MATN073
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                          MATN074
  630 JROW=INDEX(L,1)                                                   MATN075
      JCOLU =INDEX(L,2)                                                 MATN076
      DO 705 M=1,N                                                      MATN077
      SWAP=AC(M,JROW)                                                   MATN078
      AC(M,JROW)=AC(M,JCOLU)                                            MATN079
      AC(M,JCOLU)=SWAP                                                  MATN080
  705 CONTINUE                                                          MATN081
  710 CONTINUE                                                          MATN082
  740 CONTINUE                                                          MATN083
      DO 901 I=1,N                                                      MATN084
      DO 901 J=1,N                                                      MATN085
  901 AC(I,J)=AC(I,J)/1000000.                                          MATN086
C     PRINT INVERSE OF MATRIX                                         * MATN087
      IF(IPRINT)300,300,301                                             MATN088
```

```
  301 WRITE(OUT,302)                                                    MATN089
  302 FORMAT(1H1 ,//,20X,'INVERSE OF MATRIX',//)                        MATN090
      WRITE(OUT,303)(I,I=1,N)                                           MATN091
  303 FORMAT(/,12X,10I12,/)                                             MATN092
      DO 305 I=1,N                                                      MATN093
  305 WRITE(OUT,304)I,(AC(I,J),J=1,N)                                   MATN094
  304 FORMAT(/,1X,'ROW NO.',I3,1X,10E12.6,/,11(12X,10E12.6,/))          MATN095
C     COMPUTE SOLUTION VECTOR                                         * MATN096
  300 DO 750 I=1,N                                                      MATN097
      B(I)=0.0                                                          MATN098
      DO 750 J=1,N                                                      MATN099
      B(I)=B(I)+AC(I,J)*WS(J)/8.34                                      MATN100
  750 CONTINUE                                                          MATN101
C     PRINT SYSTEM SOLUTION VECTOR                                    * MATN102
      WRITE(OUT,88)(I,I=1,N1)                                           MATN103
   88 FORMAT('1',50X,'SYSTEM SOLUTION VECTOR',/, 25X,                   MATN104
     1'C O M P O N E N T   C O N C E N T R A T I O N  (MG/L)',          MATN105
     2/,'  SECTION',5X,'MILE PT',4X,8I8)                                MATN106
      DO 61 I=1, N2                                                     MATN107
   61 WRITE(OUT,101)I,PTMIL(I),(B(J),J=I,N,N2)                          MATN108
  101 FORMAT(3X,I3,7X,F6.2,6X,8F8.3)                                    MATN109
      RETURN                                                            MATN110
      END                                                               MATN111
      SUBROUTINE SENS                                                   SENS001
C     ********************************************************************** SENS002
C     ********************************************************************** SENS003
C     THIS SUBROUTINE IS USED TO VARY --  THE DECAY CONSTANTS K(II,KK) * SENS004
C     EACH SEGMENT AND/OR WASTE LOAD(S) INTO THE SYSTEM               * SENS005
C                                                                     * SENS006
C     ON CHANGES OF DECAY CONSTANT(S) THE PROGRAM ASSUMES THEY WILL BE * SENS007
C     MODIFIED FOR ALL SEGMENTS                                       * SENS008
C                                                                     * SENS009
C                                                                     * SENS010
C     ********************************************************************** SENS011
      REAL K,LOAD                                                       SENS012
      INTEGER OUT,REVIS                                                 SENS013
      COMMON/BLK1/N,NS,NP,INPUT,OUT,A(180,180)                          SENS014
      COMMON/BLK3/V(60),TEMPER(60),CHLOR(60)                            SENS015
      COMMON/BLK4/PTMIL(60),XK1(180)                                    SENS016
      COMMON/BLK5/WS(180)                                               SENS017
      COMMON/UT1/ROW(180),ROW2(180)                                     SENS018
      COMMON/UT2/K(60)                                                  SENS019
      COMMON/OPTION/IPRINT,ISENS,KWDEF,KBOD,NRUNS                       SENS020
C                                                                     * SENS021
   44 READ(INPUT,9)NRATES,NLOADS                                        SENS022
    9 FORMAT(2I2)                                                       SENS023
      IF(NLOADS.EQ.0)GO TO 20                                           SENS024
      WRITE(OUT,10)NLOADS                                               SENS025
   10 FORMAT('1',//,10X,'SYSTEM SENSITIVITY ANALYSIS  *  '              SENS026
     1 ,'NUMBER OF NEW WASTE LOADS = ',I2, //,20X,'LOAD (LB/DAY)',10X,  SENS027
     2 'SECTION',10X,'COMPONENT')                                       SENS028
      DO 41 L=1,NLOADS                                                  SENS029
C                                                                     * SENS030
C********************************************************************** SENS031
C                                                                     * SENS032
C     --  CHANGE WASTE LOADS --                                       * SENS033
C                                                                     * SENS034
C                                                                     * SENS035
C********************************************************************** SENS036
```

```
   30 READ(INPUT,31)LOAD,ISEG,IREACT                                    SENS037
   31 FORMAT(F10.0,2I2)                                                 SENS038
      IF(ISEG)32,32,22                                                  SENS039
   22 WRITE(OUT,14)LOAD,ISEG,IREACT                                     SENS040
   14 FORMAT(23X,F10.0,12X,I3,14X,I3)                                   SENS041
      IPOS=NS*(IREACT-1)+ISEG                                           SENS042
      WS(IPOS) = LOAD                                                   SENS043
   41 CONTINUE                                                          SENS044
   32 WRITE(OUT,33)                                                     SENS045
   33 FORMAT('1',45X,'REVISED SYSTEM LOAD VECTOR          '//,40X,      SENS046
     1'COMPONENT',5X,'SECTION',5X,'LOAD (LB/DAY)')                      SENS047
      II=1                                                              SENS048
      DO 35 I=1,N                                                       SENS049
      DO  35 L=1,NS                                                     SENS050
      WRITE(OUT,66)I,L,WS(II)                                           SENS051
   66 FORMAT(42X,I3,9X,I3,9X,F12.1)                                     SENS052
      II=II+1                                                           SENS053
   35 CONTINUE                                                          SENS054
      IF(NRATES.GT.0)GO TO 20                                           SENS055
C*************************************************************************SENS056
C                                                                       * SENS057
C     CHANGES WILL BE FOR WASTE VECTOR ONLY                             * SENS058
C                                                                       * SENS059
C                                                                       * SENS060
C*************************************************************************SENS061
      NRUNS=NRUNS-1                                                     SENS062
C*************************************************************************SENS063
C                                                                       * SENS064
C     MULTIPLY NEW WASTE VECTOR BY INVERSE OF SYSTEM MATRIX             * SENS065
C                                                                       * SENS066
C*************************************************************************SENS067
   37 DO 36 I=1,NP                                                      SENS068
      XK1(I)=0.0                                                        SENS069
      DO 36 L=1,NP                                                      SENS070
   36 XK1(I)=XK1(I)+A(I,L)*WS(L)/8.34                                   SENS071
C*************************************************************************SENS072
C                                                                       * SENS073
C     WRITE SOLUTION VECTOR                                             * SENS074
C                                                                       * SENS075
C*************************************************************************SENS076
      WRITE(OUT,188)(I,I=1,N)                                           SENS077
  188 FORMAT('1',50X,'SYSTEM SOLUTION VECTOR',/, 25X,                   SENS078
     1'C O M P O N E N T   C O N C E N T R A T I O N  (MG/L)',          SENS079
     2 /,3X,'SECTION',5X,'MILE PT',4X,8I8)                              SENS080
      DO 61 I=1, NS                                                     SENS081
   61 WRITE(OUT,101)I,PTMIL(I),(XK1(J),J=I,NP,NS)                       SENS082
  101 FORMAT(4X,I3,7X,F6.2,6X,8F8.3)                                    SENS083
   38 IF(KBOD.GT.0)CALL DODEF                                           SENS084
      IF(NRUNS)99,99,44                                                 SENS085
C*************************************************************************SENS086
C                                                                       * SENS087
C     CHANGES OF REACTION RATES --                                      * SENS088
C                                                                       * SENS089
C*************************************************************************SENS090
   20 IF(NRATES.LE.0)GO TO 99                                           SENS091
      WRITE(OUT,180)(I,I=1,NS)                                          SENS092
  180 FORMAT('1', 20X,'SYSTEM SENSITIVITY ANALYSIS  *   REVISED SYSTEM  SENS093
     1 REACTION RATE CONSTANTS (1/DAY)',//,25X,                         SENS094
     1'S  E  C  T  I  O  N    N  U  M  B  E  R  ',/,'  REACTION',3X,    SENS095
```

```
     2 'THETA',5X,13I8,/,4(13X,13I8,/))                                    SENS096
       DO 115 I=1,NS                                                       SENS097
       READ(1'I)ROW                                                        SENS098
       DO 29 J=1,N                                                         SENS099
       JJ=NS*(J-1)+ I                                                      SENS100
       READ(3'JJ)ROW2                                                      SENS101
       DO 60 L=1,NS                                                        SENS102
       LL = NS *(J-1)+L                                                    SENS103
C*************************************************************************SENS104
C                                                                         *SENS105
C     FORMATION OF THE LARGE MATRIX WHICH CONSISTS OF KV AND ROWMATRICES  *SENS106
C                                                                         *SENS107
C*************************************************************************SENS108
    60 ROW2(LL)=ROW2(LL)+ROW(L)                                            SENS109
       DO 93 JC=1,NP                                                       SENS110
    29 CONTINUE                                                            SENS112
   115 CONTINUE                                                            SENS113
       DO 25 I=1,NRATES                                                    SENS114
       READ(INPUT,2)II,KK,THETA                                            SENS115
     2 FORMAT(2I2,1X,F5.3)                                                 SENS116
C*************************************************************************SENS117
C                                                                         *SENS118
C     READ REVISED DECAY RATE K(II,KK) FOR ALL SEGMENTS                   *SENS119
C                                                                         *SENS120
C*************************************************************************SENS121
       IF(II.GT.0.AND.KK.GT.0)GO TO 21                                     SENS122
       WRITE(OUT,5)                                                        SENS123
     5 FORMAT('  PROGRAM TERMINATED DURING SYSTEM SENSITIVITY ANALYSIS     SENS124
      1DUE TO ERRORS IN REACTION RATES DATA INPUT' )                       SENS125
       CALL EXIT                                                           SENS126
    21 READ(INPUT,3)(K(L),L=1,NS)                                          SENS127
     3 FORMAT(8F10.0)                                                      SENS128
C*************************************************************************SENS129
C                                                                         *SENS130
C     TEMPERATURE CORRECT NEW DECAY CONSTANTS                             *SENS131
C                                                                         *SENS132
C*************************************************************************SENS133
       IF(THETA.LE.0)GO TO 82                                              SENS134
       DO 83  L=1,NS                                                       SENS135
    83 K(L)=K(L)*THETA**(TEMPER(L)-20)                                     SENS136
    82 WRITE(OUT,4)II,KK,THETA,(K(L),L=1,NS)                               SENS137
     4 FORMAT(I3,' TO ',I3,3X,F5.3,5X,13F8.4,/,4(13X,I3,F8.4,/))           SENS138
       ICOL =(II-1)*NS+1                                                   SENS139
       IROW = (KK-1)*NS+1                                                  SENS140
       DO 26 ISEG=1,NS                                                     SENS141
       A(IROW,ICOL)=V(ISEG)*K(ISEG)                                        SENS142
C*************************************************************************SENS143
C                                                                         *SENS144
C     ADD NEW V * K TERM TO SYSTEM MATRIX                                 *SENS145
C                                                                         *SENS146
C*************************************************************************SENS147
       IF(II-KK)11,112,11                                                  SENS148
    93 A(JJ,JC)=ROW2(JC)                                                   SENS111
   112 READ(1'ISEG)(ROW(L),L=1,NS)                                         SENS149
       A(IROW,ICOL)=A(IROW,ICOL)+ROW(ISEG)                                 SENS150
       GO TO 13                                                            SENS151
    11 A(IROW,ICOL)=-A(IROW,ICOL)                                          SENS152
C*************************************************************************SENS153
C                                                                         *SENS154
```

```
C      NOTE  --  PROGRAM ASSUMES CHANGES ARE MADE FROM *** ORIGINAL *** * SENS155
C                SYSTEM MATRIX, IF CUMMULATIVE EFFECTS ARE DESIRED      * SENS156
C                SAVE THIS NEW SYSTEM MATRIX ON OFFLINE DISK FILE 1     * SENS157
C                                                                       * SENS158
C*********************************************************************** SENS159
                                                                          SENS160
   13 IROW=IROW+1                                                         SENS161
      ICOL=ICOL+1                                                         SENS162
   26 CONTINUE                                                            SENS163
   25 CONTINUE                                                            SENS164
      NRUNS=NRUNS-1                                                       SENS165
      IF(IPRINT)90,90,70                                                  SENS166
   70 WRITE(OUT,80)                                                       SENS167
   80 FORMAT(1H1,///,43X,'REVISED SYSTEM MATRIX BEFORE SOLUTION',///)     SENS168
      WRITE(OUT,89)(I,I=1,NP)                                             SENS169
   89 FORMAT(/,12X,10I12,/)                                               SENS170
      DO 39 IR=1,NP                                                       SENS171
      WRITE(OUT,12)IR,(A(IR,JC),JC=1,NP)                                  SENS172
   12 FORMAT(1X,'ROW NO.',I3,1X,10F12.2/(12X,10F12.2))                    SENS173
   39 CONTINUE                                                            SENS174
   90 RETURN                                                              SENS175
   99 CALL EXIT                                                           SENS176
      END                                                                 DODEF001
      SUBROUTINE DODEF                                                   *DODEF002
C                                                                        *DODEF003
C     ***************************************************************** *DODEF004
C     THIS SUBROUTINE WILL COMPUTE DISSOLVED OXYGEN DEFICIT             *DODEF005
C     BY SELECTING COMPONENTS CAUSING DEFICIT WITHIN THE SYSTEM         *DODEF006
C                                                                       *DODEF007
C     THE OUTPUT WILL BE D.O. DEFICIT DUE TO EACH COMPONENT AND         *DODEF008
C     TOTAL DEFICIT DUE TO ALL COMPONENTS IN EACH SECTION OF THE SYSTEM *DODEF009
C     ***************************************************************** *DODEF010
C                                                                        DODEF011
      INTEGER OUT                                                        DODEF012
      REAL KA                                                            DODEF013
      DIMENSION STOCH(10),ICOEF1(10),ICOEF2(10),BMAT(60,60),CS(60)       DODEF014
      DIMENSION VK(60),RES(60),PROD(60),DODSUM(60),SROW(60),WDODEF(60)    DODEF015
      COMMON/UT1/IPIVO(180),PIVOT(180)                                   DODEF016
      COMMON/UT2/INDEX(180,2),KA(60)                                     DODEF017
      COMMON/BLK1/NR,N,NP,INPUT,OUT                                      DODEF018
      COMMON/BLK3/V(60),TEMPER(60),CHLOR(60)                             DODEF019
      COMMON/BLK4/PTMIL(60),XK1(180)                                     DODEF020
      COMMON/OPTION/IPRINT,ISENS,KWDEF                                   DODEF021
      EQUIVALENCE (CS(1),CHLOR(1)),(KA(1),WDODEF(1))                     DODEF022
      EQUIVALENCE (SROW(1),PIVOT(1))                                     DODEF023
      DO 1  I=1,N                                                        DODEF024
      DODSUM(I)=0.0                                                      DODEF025
      RES(I)=0.0                                                         DODEF026
      PROD(I)=0.0                                                        DODEF027
      VK(I)=0.0                                                          DODEF028
    1 CONTINUE                                                           DODEF029
      IF(IDODSW.EQ.999)GO TO 30                                         *DODEF030
C*********************************************************************** *DODEF031
C                                                                        *DODEF032
C     READ REAERATION RATES                                              *DODEF033
C                                                                        *DODEF034
C*********************************************************************** DODEF035
      READ(INPUT,10)(KA(I),I=1,N)                                        DODEF036
   10 FORMAT(16F5.3)                                                    *DODEF037
C*********************************************************************** 
```

```
C                                                                      *DODEF038
C     IF KA ARE SAME IN ALL SECTIONS, INPUT ONLY VALUE OF FIRST SECTION*DODEF039
C                                                                      *DODEF040
C***********************************************************************DODEF041
      IF(KA(2))2,2,3                                                    DODEF042
    2 DO 4  I=2,N                                                       DODEF043
      KA(I)=KA(1)                                                       DODEF044
    4 CONTINUE                                                          DODEF045
C***********************************************************************DODEF046
C                                                                      *DODEF047
C     TEMPERATURE CORRECT REAERATION RATES                             *DODEF048
C***********************************************************************DODEF050
      DO 6  I=1,N                                                       DODEF051
      KA(I)=KA(I)*1.024**(TEMPER(I)-20.)                                DODEF052
    6 CONTINUE                                                          DODEF053
C***********************************************************************DODEF054
C                                                                      *DODEF055
C     PRINT REAERATION RATES                                           *DODEF056
C                                                                      *DODEF049
C                                                                      *DODEF057
C***********************************************************************DODEF058
    3 WRITE(OUT,8)                                                      DODEF059
    8 FORMAT('1',40X,'DISSOLVED OXYGEN DEFICIT ANALYSIS',//,5X,         DODEF060
     1'SEGMENT',10X,'TEMPERATURE (CENT.)',8X,'REAERATION RATE (1/DAY)' )DODEF061
      WRITE(OUT,9)(I,TEMPER(I),KA(I),I=1,N)                             DODEF062
    9 FORMAT(5X,I4,14X,F6.2,21X,F12.6)                                  DODEF063
C***********************************************************************DODEF064
C                                                                      *DODEF065
C     COMPUTE SATURATION LEVEL OF DISSOLVED OXYGEN IN EA. SECTION      *DODEF066
C                                                                      *DODEF067
C***********************************************************************DODEF068
      DO 7  I=1,N                                                       DODEF069
      CS(I)=(1.-.000009*CHLOR(I))*(14.652-.41022*TEMPER(I)+             DODEF070
     1.0079910*TEMPER(I)**2.-.000077774*TEMPER(I)**3.)                  DODEF071
    7 CONTINUE                                                          DODEF072
C     COMPUTE DEFICIT MATRIX                                           *DODEF073
      DO 101 I=1,N                                                      DODEF074
      READ(1'I)(BMAT(I,J),J=1,N)                                        DODEF075
      BMAT(I,I)=BMAT(I,I)+V(I)*KA(I)                                    DODEF076
  101 CONTINUE                                                          DODEF077
C***********************************************************************DODEF078
C                                                                      *DODEF079
C     PRINT DISS. OXYGEN DEF. MATRIX IF DESIRED                        *DODEF080
C                                                                      *DODEF081
C***********************************************************************DODEF082
      IF(IPRINT)90,90,70                                                DODEF083
   70 WRITE(OUT,280)                                                    DODEF084
  280 FORMAT(1H1,///,50X,'DEFICIT MATRIX BEFORE INVERSION',///)         DODEF085
      WRITE(OUT,89)(I,I=1,N)                                            DODEF086
   89 FORMAT(/,12X,10I12,/)                                             DODEF087
      DO 39 IR=1,N                                                      DODEF088
      WRITE(OUT,162)IR,(BMAT(IR,JC),JC=1,N)                             DODEF089
  162 FORMAT(1X,'ROW NO.',I3,1X,10E12.6/(12X,10E12.6))                  DODEF090
   39 CONTINUE                                                          DODEF091
C***********************************************************************DODEF092
C                                                                      *DODEF093
C     COMPUTE INVERSE OF DEFICIT MATRIX                                *DODEF094
C                                                                      *DODEF095
C***********************************************************************DODEF096
```

```
   90 DO 900  I=1,N                                                    DODEF097
      DO 900  J=1,N                                                    DODEF098
  900 BMAT(I,J)=BMAT(I,J)/1000000.                                     DODEF099
      DO 20 J=1,N                                                      DODEF100
      IPIVO(J)=0                                                       DODEF101
      DO 21 I=1,2                                                      DODEF102
   21 INDEX(J,I)=0                                                     DODEF103
   20 CONTINUE                                                         DODEF104
      DO 550 I=1,N                                                     DODEF105
C                                                                     *DODEF106
C     SEARCH FOR PIVOT ELEMENT                                        *DODEF107
C                                                                     *DODEF108
      AMAX=0.0                                                         DODEF109
      DO 105 J=1,N                                                     DODEF110
      IF (IPIVO (J)-1) 60, 105, 60                                     DODEF111
   60 DO 100 M=1,N                                                     DODEF112
      IF (IPIVO (M)-1) 80, 100, 740                                    DODEF113
   80 IF (ABS(AMAX)-ABS(BMAT(J,M)))85, 100, 100                        DODEF114
   85 IROW=J                                                           DODEF115
      ICOLU =M                                                         DODEF116
      AMAX=BMAT(J,M)                                                   DODEF117
  100 CONTINUE                                                         DODEF118
  105 CONTINUE                                                         DODEF119
      IPIVO (ICOLU )=IPIVO (ICOLU )+1                                  DODEF120
C                                                                     *DODEF121
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL               *DODEF122
C                                                                     *DODEF123
      IF (IROW-ICOLU ) 150, 260, 150                                   DODEF124
  150 DO 200 L=1,N                                                     DODEF125
      SWAP=BMAT(IROW,L)                                                DODEF126
      BMAT(IROW,L)=BMAT(ICOLU,L)                                       DODEF127
  200 BMAT(ICOLU,L)=SWAP                                               DODEF128
  260 INDEX(I,1)=IROW                                                  DODEF129
      INDEX(I,2)=ICOLU                                                 DODEF130
      PIVOT(I)=BMAT(ICOLU,ICOLU)                                       DODEF131
C                                                                     *DODEF132
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                               *DODEF133
C                                                                     *DODEF134
      BMAT(ICOLU,ICOLU)=1.0                                            DODEF135
      DO 350 L=1,N                                                     DODEF136
  350 BMAT(ICOLU,L)=BMAT(ICOLU,L)/PIVOT(I)                             DODEF137
C                                                                     *DODEF138
C     REDUCE NON-PIVOT ROWS                                           *DODEF139
C                                                                     *DODEF140
      DO 550 L1=1,N                                                    DODEF141
      IF(L1-ICOLU ) 400, 550, 400                                      DODEF142
  400 S=BMAT(L1,ICOLU)                                                 DODEF143
      BMAT(L1,ICOLU)=0.0                                               DODEF144
      DO 450 L=1,N                                                     DODEF145
  450 BMAT(L1,L)=BMAT(L1,L)-BMAT(ICOLU,L)*S                            DODEF146
  550 CONTINUE                                                         DODEF147
C                                                                     *DODEF148
C     INTERCHANGE COLUMNS                                             *DODEF149
C                                                                     *DODEF150
      DO 710 I=1,N                                                     DODEF151
      L=N+1-I                                                          DODEF152
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                         DODEF153
  630 JROW=INDEX(L,1)                                                  DODEF154
      JCOLU =INDEX(L,2)                                                DODEF155
```

```
      DO 705 M=1,N                                                        DODEF156
      SWAP=BMAT(M,JROW)                                                   DODEF157
      BMAT(M,JROW)=BMAT(M,JCOLU)                                          DODEF158
      BMAT(M,JCOLU)=SWAP                                                  DODEF159
  705 CONTINUE                                                            DODEF160
  710 CONTINUE                                                            DODEF161
  740 CONTINUE                                                            DODEF162
      DO 901 I=1,N                                                        DODEF163
      DO 901 J=1,N                                                        DODEF164
  901 BMAT(I,J)=BMAT(I,J)/1000000.                                        DODEF165
      IODSW=999                                                           DODEF166
C*************************************************************************DODEF167
C                                                                        *DODEF168
C     PRINT INVERSE OF DEFICIT MARTIX IF DESIRED                         *DODEF169
C                                                                        *DODEF170
C*************************************************************************DODEF171
      IF(IPRINT)139,139,370                                               DODEF172
  370 WRITE(OUT,281)                                                      DODEF173
  281 FORMAT(1H1,//,40X,'INVERSE OF DEFICIT MATRIX',///)                  DODEF174
      WRITE(OUT,89)(I,I=1,N)                                              DODEF175
      DO 339 IR=1,N                                                       DODEF176
      WRITE(OUT,162)IR,(BMAT(IR,JC),JC=1,N)                               DODEF177
  339 CONTINUE                                                            DODEF178
C*************************************************************************DODEF179
C                                                                        *DODEF180
C     READ IN DEFICIT IN LB/DAY DUE TO BENTHAL DEMAND AND DIRECT         *DODEF181
C     DISCHARGES INTO THE SECTION                                        *DODEF182
C                                                                        *DODEF183
C*************************************************************************DODEF184
  139 IF(KWDEF.EQ.0)GO TO 439                                             DODEF185
      READ(INPUT,16)(WDODEF(I),I=1,N)                                     DODEF186
   16 FORMAT(8F10.0)                                                      DODEF187
      WRITE(OUT,17)                                                       DODEF188
   17 FORMAT('1',///,20X,'DEFICIT IN SYSTEM DUE TO BACKGROUND LOADS',     DODEF189
     1 //,5X,'SECTION',5X,'MILE PT',5X,'DEFICIT CONCENTRATION (MG/L)',5XDODEF190
     1 ,'D.O. SATURATION (MG/L)',5X,'DISSOLVED OXYGEN (MG/L)' )           DODEF191
      DO 110 I=1,N                                                        DODEF192
      DODSUM(I)=0.0                                                       DODEF193
      DO 111 J=1,N                                                        DODEF194
  111 DODSUM(I)=DODSUM(I)+BMAT(I,J)*WDODEF(J)                             DODEF195
      DO = CS(I)-DODSUM(I)                                                DODEF196
      WRITE(OUT,13)I,PTMIL(I),DODSUM(I),CS(I),DO                          DODEF197
  110 CONTINUE                                                            DODEF198
C*************************************************************************DODEF199
C                                                                        *DODEF200
C     READ NUMBER OF REACTIONS AND REACTION SCHEMES PRODUCING DEFICIT    *DODEF201
C                                                                        *DODEF202
C*************************************************************************DODEF203
  439 READ(INPUT,11)NRDEF,(ICOEF1(J),ICOEF2(J),STOCH(J),J=1,NRDEF)        DODEF204
   11 FORMAT(I5,5X,7(2I2,F6.0))                                           DODEF205
C                                                                        *DODEF206
   30 DO 103  II=1,NRDEF                                                  DODEF207
C*************************************************************************DODEF208
C                                                                        *DODEF209
C        CONSTRUCT V(ISEG)*K(I1,I2) VECTOR FOR EA REACTION               *DODEF210
C        PRODUCING DEFICIT FROM LARGE VK SYSTEM MATRIX ON OFFLINE DISK   *DODEF211
C                                                                        *DODEF212
C*************************************************************************DODEF213
      IROW=(ICOEF2(II)-1)*N+1                                             DODEF214
```

```
      ICOL=(ICOEF1(II)-1)*N+1                                          DODEF215
      LAST=IROW+N-1                                                    DODEF216
      ISTRT=1                                                          DODEF217
      DO 104 J=IROW,LAST                                               DODEF218
      READ(3'J)(SROW(K),K=1,NP)                                        DODEF219
      VK(ISTRT)=ABS(SROW(ICOL))                                        DODEF220
      ISTRT=ISTRT+1                                                    DODEF221
      ICOL=ICOL+1                                                      DODEF222
  104 CONTINUE                                                         DODEF223
C*************************************************************************DODEF224
C                                                                      *DODEF225
C     MULTIPLY  (VK)*(STOCH COEFF)*(SOLUTION VECTOR)                   *DODEF226
C                                                                      *DODEF227
C*************************************************************************DODEF228
      LCC=N*(ICOEF1(II)-1)+1                                           DODEF229
      DO 155  I=1,N                                                    DODEF230
      RES(I)=XK1(LCC)*STOCH(II)*VK(I)                                  DODEF231
      LCC=LCC+1                                                        DODEF232
  155 CONTINUE                                                         DODEF233
C*************************************************************************DODEF234
C                                                                      *DODEF235
C     MULTIPLY  (VK)*STOCH*(SOL) BY INVERSE OF DEFICIT MATRIX          *DODEF236
C                                                                      *DODEF237
C*************************************************************************DODEF238
      DO 106  I=1,N                                                    DODEF239
      PROD(I)=0.0                                                      DODEF240
      DO 106  J=1,N                                                    DODEF241
      PROD(I)=PROD(I)+BMAT(I,J)*RES(J)                                 DODEF242
  106 CONTINUE                                                         DODEF243
C*************************************************************************DODEF244
C                                                                      *DODEF245
C     PRINT RESULTING DEFICIT FOR REACTION                             *DODEF246
C                                                                      *DODEF247
C*************************************************************************DODEF248
      WRITE(OUT,12)ICOEF1(II),ICOEF2(II),STOCH(II)                     DODEF249
   12 FORMAT(//,20X,'DEFICIT DUE TO REACTION',I2,' TO ',I2,            DODEF250
     1 10X,'STOCHIOMETRIC COEFFICIENT OF ',F6.2,//,5X,                 DODEF251
     1 'SECTION',5X,'MILE PT',5X,'DEFICIT CONCENTRATION (MG/L)',5X,    DODEF252
     1 'D.O. SATURATION (MG/L)',5X,'DISSOLVED OXYGEN (MG/L)' )         DODEF253
      DO 108  I=1,N                                                    DODEF254
      DO=CS(I)-PROD(I)                                                 DODEF255
      WRITE(OUT,13)I,PTMIL(I),PROD(I),CS(I),DO                         DODEF256
  108 CONTINUE                                                         DODEF257
   13 FORMAT(7X,I3,8X,F5.1,13X,F7.3,25X,F7.3,19X,F7.3)                 DODEF258
C*************************************************************************DODEF259
C                                                                      *DODEF260
C     SUM REACTION DEFICIT FOR TOTAL DEFICIT COMPUTATION               *DODEF261
C                                                                      *DODEF262
C*************************************************************************DODEF263
      DO 107  I=1,N                                                    DODEF264
      DODSUM(I)=DODSUM(I)+PROD(I)                                      DODEF265
  107 CONTINUE                                                         DODEF266
  103 CONTINUE                                                         DODEF267
C*************************************************************************DODEF268
C                                                                      *DODEF269
C     PRINT TOTAL DISSOLVED OXYGEN DEFICIT                             *DODEF270
C                                                                      *DODEF271
C*************************************************************************DODEF272
      WRITE(OUT,14)(ICOEF1(I),ICOEF2(I), I=1,NRDEF)                    DODEF273
```

```
      WRITE(OUT,15)                                                          DODEF274
   14 FORMAT('1',40X,'TOTAL DISSOLVED OXYGEN DEFICIT FOR REACTIONS',         DODEF275
     1 10(/,76X,I2,' TO ',I2))                                               DODEF276
   15 FORMAT(///,5X,'SECTION',5X,'MILE PT',5X,                               DODEF277
     2'TOTAL DEFICIT CONCENTRATION (MG/L)',5X,'DISSOLVED OXYGEN (MG/L)')     DODEF278
      DO 109  I=1,N                                                          DODEF279
      DO=CS(I)-DODSUM(I)                                                     DODEF280
      WRITE(OUT,13)I,PTMIL(I),DODSUM(I),DO                                   DODEF281
  109 CONTINUE                                                               DODEF282
      RETURN                                                                 DODEF283
      END                                                                    DODEF284
```

INPUT DESCRIPTION

1. TITLE CARD

COLUMNS	VARIABLE	FORMAT	DESCRIPTION
1-80	Title	20A4	Users description of run

2. SYSTEM SIZE CARD

COLUMNS	VARIABLE	FORMAT	DESCRIPTION
1- 2	N	I2	Number of system segments
3- 4	NR	I2	Number of components (reactants)

3. PROGRAM OPTIONS - If option desired, enter a positive integer in corresponding variable.

COLUMNS	VARIABLE	FORMAT	DESCRIPTION
1- 2	IPRINT	I2	Print system matrix and its inverse
3- 4	ISENS	I2	Perform sensitivity analysis of system
5- 6	KBOD	I2	Perform dissolved oxygen deficit analysis
7- 8	KWDEF	I2	Control variable to read direct (background) sources of deficit in the deficit analysis subroutine

4. SYSTEM GEOMETRY CONTROL CARD

COLUMNS	VARIABLE	FORMAT	DESCRIPTION
1- 7	SCALE (1)	F7.0	Area scale factor ft^2/ou*
8-14	SCALE (2)	F7.0	Dispersion scale factor mi^2/day/ou
15-21	SCALE (3)	F7.0	Flow scale factor CFS/ou
22-28	SCALE (4)	F7.0	Length scale factor ft/ou

*note: ou stands for the "original units" in which the parameter to be converted by the scale factor is expressed. The purpose of the scale factors is to allow the user to input parameters in units which are different from those specified on the following pages. For instance, according to this program length should be input as feet. However, it may be more convenient to enter it in miles and set (SCALE(4) to 5280. The program would then internally convert the length from miles to feet

5. INTERFACE PARAMETER CARDS (2 per segment; total number = 2*N)

Columns	Variable	Format	Description	Units
1- 6	AREA	F6.0	Cross sectional area of interface with section	ft^2
7-12	E	F6.0	Dispersion coefficient of first interface with section	mi^2/day
13-18	Q	F6.0	Flow * across the first interface with section	cfs
19-21	IARAY	I3	Section forming the first interface with section	
22-27	AREA	F6.0	Cross sectional area of second interface	ft^2
28-33	E	F6.0	Dispersion coefficient of second interface	mi^2/day
34-39	Q	F6.0	Flow across the second interface	cfs
40-42	IARAY	I3	Section forming the second interface	
43-48	AREA	F6.0	Cross sectional area of third interface	ft^2
49-54	E	F6.0	Dispersion coefficient of third interface	mi^2/day
55-60	Q	F6.0	Flow across the third interface	cfs
61-63	IARAY	I3	Section forming the third interface	

Note: The second interface parameter card is identical to above, but for the fourth, fifth and sixth interfaces. If the segment in question has an interface which forms a boundary across which mass is transported (that is, a boundary condition) interface parameters must be input for it. To do this, input the appropriate AREA, E and Q for the interface and input the segment number as the IARAY.

It is only necessary to input the parameters for a particular interface once. In other words after inputting the parameters of the interface of segment 1 with segment 2, it is not necessary to input the parameters of the interface of segment 2 with segment 1.

* flow out of a segment is positive; flow into a segment is negative.

6. CHARACTERISTIC LENGTHS (1 card per section; 1 length for each interface; therefore up to 6 lengths per card)

Columns	Variable	Format	Description	Units
1- 3	NLA(I,JI)	I3	First segment which forms an interface with segment I	
4-13	LA(I,JI)	F10.0	Length of segment I with respect to section JI*	feet
14-16	NLA(I,JJ)	I3	Second segment which forms an interface with segment I.	
17-26	LA(I,JJ)	F10.0	Length of segment I with respect to segment JJ	feet
27-29	NLA(I,JK)	I3	Third segment which forms an interface with segment I	
30-39	LA(I,JK)	F10.0	Length of segment I with respect to segment JK	feet
40-42	NLA(I,JL)	I3	Fourth segment which forms an interface with segment I	
43-52	LA(I,JL)	F10.0	Length of segment I with respect to segment JL	feet
53-55	NLA(I,JM)	I3	Fifth segment which forms an interface with segment I	
56-65	LA(I,JM)	F10.0	Length of segment I with respect to segment JM	feet
66-68	NLA(I,JN)	I3	Sixth segment which forms an interface with segment I	
69-78	LA(I,JN)	F10.0	Length of segment I with respect to segment JN	feet

* note: the following figure illustrates the interpretation of characteristic lengths as used in this program.

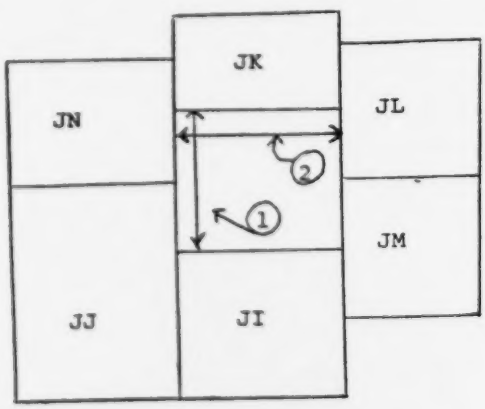


Figure 7: Depiction of characteristic lengths

Where

① - LA(I,JK) and LA(I,JI)

② - LA(I,JJ), LA(I,JM) LA(I,JL) and LA(I,JN)

As can be seen, any other than orthogonally shaped segments would present a problem as to the measurement of LA. It is therefore reiterated that oddly shaped segmentation schemes should be avoided except when absolutely necessary to depict the natural geometry of the system.

Note also that for sections which form boundaries across which mass is transported, the characteristic length of the segment with respect to the boundary must be inputted. This length is read in as LA(I,I), where I is the boundary section. (see equations 23 and 25).

7. SECTION CHLORIDE CONCENTRATIONS. Optional. This data is only necessary if KBOD>1 (see card type #3)

Columns	Variable	Format	Description
1-10	CHLOR(1)	F10.0	Chloride concentration (mg/l) at the 1st section
11-20	CHLOR(2)	F10.0	Chloride concentration at the second section
.	.	.	
.	.	.	
.	.	.	
71-80	CHLOR(8)	F10.0	Chloride concentration (mg/l) at the 8th section

The card above is repeated as necessary until all chloride concentrations for the system have been entered.

8. TEMPERATURE/VOLUME CARDS

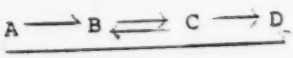
Columns	Variable	Format	Description
1-10	TEMPER(1)	F10.0	Water temperature (^{o}C) of 1st section
11-20	VOL(1)	F10.0	Volume (10^{6} ft^{3}) of 1st section
21-30	TEMPER(2)	F10.0	Water temperature (^{o}C) of 2nd section
31-40	VOL(2)	F10.0	Volume (10^{6} ft^{3}) of 2nd section
.	.	.	
.	.	.	
.	.	.	
71-80	VOL(4)	F10.0	Volume of 4th section

This card is repeated until the temperature and volume for all sections have been entered. If the temperature is constant in the system, it is only necessary to enter this parameter for the 1st section, as the program will then test for the 2nd section temperature and since it is blank it will propagate the value of the 1st section to all the sections in the system.

REACTION RATE COEFFICIENTS

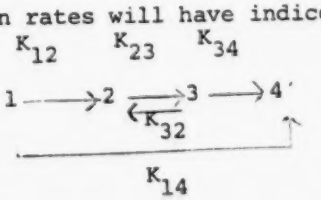
9. REACTION SCHEME CARD.

Consider the reactions:



These reactants are represented numerically in the program as follows:

The reaction rates will have indices referring to these reactant numbers:



The program will read these reaction indices as follows:

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1- 2	I	I2	1st reaction index of reaction rate constant Kij
3- 4	J	I2	2nd reaction index of reaction rate constant Kij
6-10	THETA	F 5.3	temperature correction coefficient; the program assumes the rate constants are given at 20°C and can be converted to any temperature by the equation: Kij(T°C) = Kij(20°C) * THETA**(TEMP-20). If this field is left blank, the program will not correct the reaction rates for temperature.

10. REACTION RATE CONSTANTS (Units: 1/day)

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-10	K(1)	F10.0	reaction rate constant for 1st segment
11-20	K(2)	F10.0	reaction rate constant for 2nd segment
21-30	K(3)	F10.0	reaction rate constant for 3rd segment
.	.	.	
.	.	.	
.	.	.	
71-80	K(8)	F10.0	reaction rate constant for 8th segment

The card image above is repeated until the reaction rate constant Kij; where i, j are the indices specified on card #9, have been read. The reaction rates for a new reaction are then entered by preparing a new #9 card with the appropriate indices for the new reaction, then reaction rate cards follow with the reaction rate constant for all segments.

This procedure is repeated until the reaction rate coefficients have been read for all reaction schemes. A blank card is inserted after the last set of reaction coefficients, which signals an end to this type of input.

11. DIRECT DISCHARGES

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-10	LOAD	F10.0	Waste load into segment (lb/day)
11-12	ISEC	I2	segment number where waste load is being discharged
13-14	IREAC	I2	component (number) making up waste load

These cards are repeated for all point sources into the system, the input is terminated by a blank card.

12. BOUNDARY CONDITION

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-10	BC	F10.0	Boundary concentration (mg/l)
11-12	ISYS	I2	reaction component number
13-14	ISECT	I2	section number

These cards are repeated for each reaction component at all boundary segments. The last card will again be a blank card to terminate this type of input.

13. MILEAGE POINT OF SEGMENTATION

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-10	PTMIL(4)	F10.0	Mile point designation for interface at end of first segment
11-20	PTMIL(2)	F10.	mile point at end of second segment
.	.	.	
.	.	.	
.	.	.	
71-80	PTMIL(8)	F10.	mile point at end of eighth segment

These cards are again repeated if necessary to input remaining mile points. The program will read the same number of mile points as the number of segments defined for the system on card image #2.

This mile point is used only to locate the final concentrations in the system; it will not be used for any numerical computation such as the length of a segment.

DISSOLVED OXYGEN DEFICIT ANALYSIS (KBOD>0 ON OPTION CARD)

14. SEGMENT REAERATION RATE

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1- 5	KA(1)	F5.3	Reaeration rate (1/day) for first segment
6-10	KA(2)	F5.3	Reaeration rate for second segment
.	.	.	
.	.	.	
.	.	.	
76-80	KA(16)	F5.3	Reaeration rate for sixteenth segment

This card image is repeated if necessary to read reaeration rate constants for remaining segments.

15. BACKGROUND DEFICIT DUE TO BACKGROUND SOURCES (KWDEF > 0 ON OPTION CARD)

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-10	WDODEF(1)	F10.0	Direct discharge of deficit (lb/day) into the 1st segment
11-20	WDODEF(2)	F10.0	Direct discharge of deficit (lb/day) into the 2nd segment
.	.	.	
.	.	.	
.	.	.	
71-80	WDODEF(8)	F10.0	Direct discharge of deficit (lb/day) into the 8th segment

The above card is repeated until these direct discharges have been defined for the whole system.

16. SELECTING REACTION SCHEMES PRODUCING DEFICIT

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1- 5	NRDEF	I5	Number of reactions producing deficit in the system
11-12	ICOEF1(1)	I2	1st index of reaction rate constant K_{IJ}
13-14	ICOEF2(1)	I2	2nd index of reaction rate constant K_{IJ}
15-20	STOCH(1)	F6.0	Stochiometric coefficient used in converting component concentration to equivalent oxygen utilized

The subscripting of the "variable names" in the above cards refers to the first reaction producing deficit, the same procedure is utilized for all other deficit producing reactions:

21-22	ICOEF1(2)	I2	1st index of reaction rate constant K_{IJ}
23-24	ICOEF2(2)	I2	2nd index of reaction rate constant K_{IJ}
25-30	STOCH(2)	F6.0	Stochiometric coefficient for second reaction
.	.	.	
.	.	.	
.			
71-72	ICOEF1(7)	I2	1st index of reaction rate constant K_{IJ}
73-74	ICOEF2(7)	I2	2nd index of reaction rate constant K_{IJ}
75-80	STOCH(7)	F6.0	Stochiometric coefficient for seventh reaction

NOTE: The reaeration rates constants and the reaction schemes producing deficit are read only once in the program. During consecutive runs; such as in system sensitivity analysis, there is an internal switch in the program to bypass this input and also the computation of the system deficit matrix inverse after the initial use of the DODEF subroutine.

SYSTEM SENSITIVITY ANALYSIS

(ISENS>0 ON OPTION CARD)

17. MAIN PROGRAM CONTROL VARIABLE

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-2	NRUNS	I2	Number of sensitivity analysis runs to be performed

18. SUBROUTINE CONTROL VARIABLE

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-2	NRATES	I2	Number of reaction rate schemes to be revised
3-4	NLOADS	I2	Number of loads in system source vector to be changed

19. CHANGES TO WASTE LOADS (NLOADS > 0)

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-10	LOAD	F10.0	New waste load (lb/day) into segment
11-12	ISEG	I2	Segment number of modified waste load
13-14	IREACT	I2	Component number of waste load

This card image is repeated until the number of new waste loads specified on card #18 as NLOADS have been read.

CHANGES TO REACTION RATE COEFFICIENTS (NRATES > 0)

20. REACTION RATE SCHEME TO BE MODIFIED

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1- 2	II	I2	1st index of reaction rate $K_{\underline{11},KK}$
3- 4	KK	I2	2nd index of reaction rate $K_{11,\underline{KK}}$
6-10	THETA	F5.3	Reaction rate temperature correction coefficient

21. NEW REACTION RATE CONSTANTS

COLUMNS	VARIABLE NAME	FORMAT	DESCRIPTION
1-10	K(1)	F10.0	New reaction rate constant $K_{11,KK}$ (see above) for 1st segment.
11-20	K(2)	F10.0	New reaction rate constant $K_{11,KK}$ for 2nd segment
.	.	.	
.	.	.	
.	.	.	
71-80	K(8)	F10.0	New reaction rate constant $K_{11,KK}$ for 8th segment.

This card image is repeated until the reaction rate constant specified in card #20 as $K_{11,\ KK}$ has been read in for <u>all</u> segments.

The card image #20 and the sequence of card image #20 is repeated until the number reaction schemes specified on card #19 (NRATES) has been satisfied.

A new sensitivity run (if NRUNS > 1) would begin by reading a new card image #18 (see schematic diagram for various inputs).

RESTRICTIONS

1.) FEDBAKØ3 as presently written is limited to a system of sixty sections.

2.) Each section can have a maximum of six interfaces.

3.) Only one interface of a particular section may act as a boundary.

4.) The maximum number of reactants is a variable, which when multiplied by the number of sections, it cannot exceed 180.

Restrictions (1) and (4) can be easily expanded by minor modifications to the computer program. The application of this model to the nitrogen cycle by assuming first order kinetics of the bacterial reactions is a major assumption which should be verified by laboratory studies.

APPLICATION TO THE DELAWARE ESTUARY

The area modelled in the Delaware Estuary is the 86 mile stretch of tidal river between Trenton, New Jersey and Liston Point, Delaware. Figure 8. shows this area and the corresponding system segmentation which was followed.

This area was extensively investigated during the 1960's by the Delaware Estuary Comprehensive Study. The purpose of this study was to determine the water quality objectives for the estuary, the cost and benefits of these objectives, and the treatment levels necessary to achieve these objectives. The survey data used in this test case was gathered by Hydroscience Inc. [1] for the Delaware River Basin Commission. The intent of this section is to demonstrate the user of this model its application by re-verifying the results published in the forementioned report. Interested readers are suggested to review the Hydroscience report for an in depth discussion on the rational for the reaction coefficients, as well as the differences between computed and observed values.

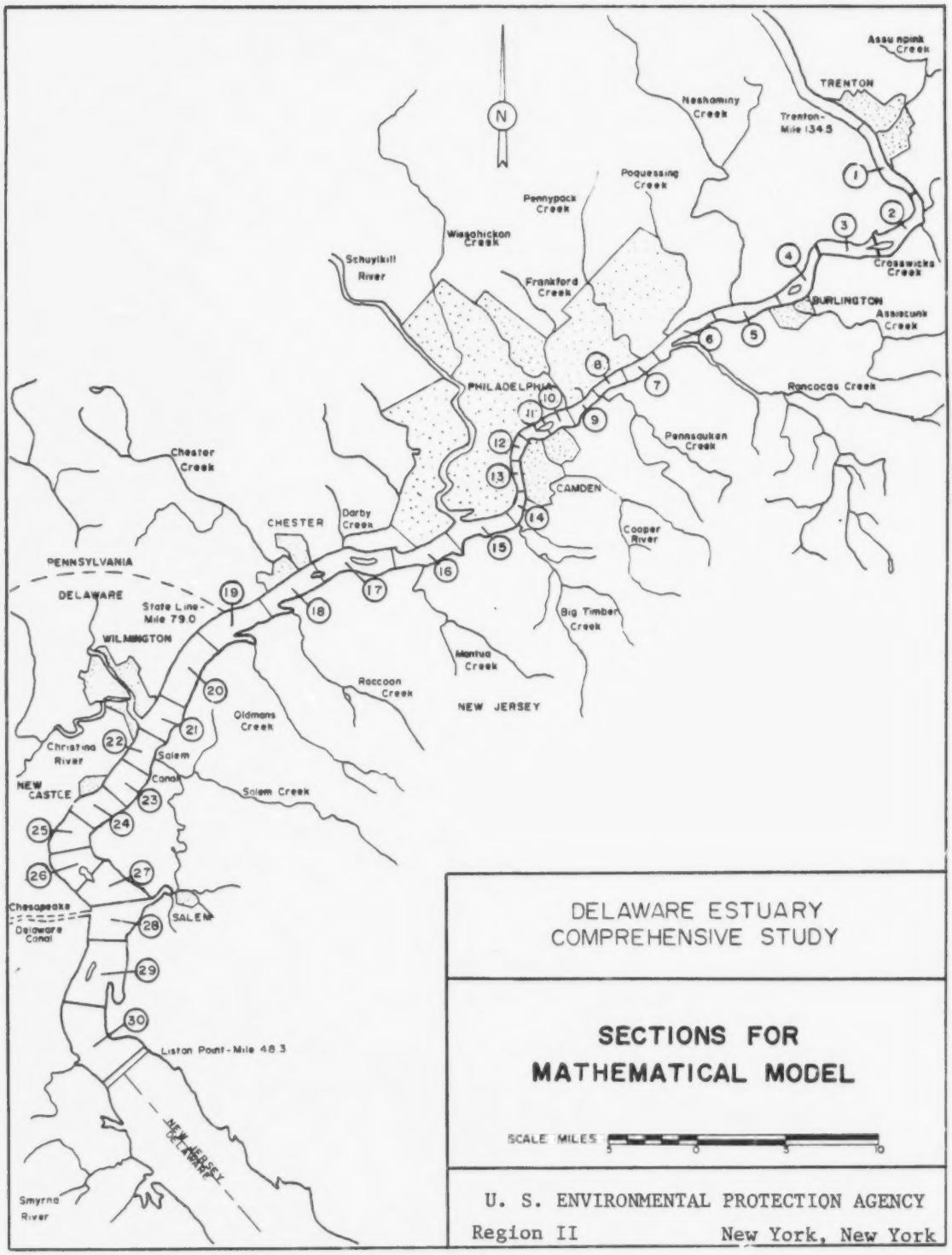

N
Assunpink Creek
TRENTON
Neshaminy Creek
Trenton-Mile 134.3
Poquessing Creek
Pennypack Creek
Wissahickon Creek
Schuylkill River
Frankford Creek
Crosswicks Creek
BURLINGTON
Assiscunk Creek
PHILADELPHIA
Rancocas Creek
Pennsauken Creek
Chester Creek
CHESTER
Darby Creek
CAMDEN
Cooper River
PENNSYLVANIA
DELAWARE
State Line-Mile 79.0
WILMINGTON
Big Timber Creek
Mantua Creek
Raccoon Creek
NEW JERSEY
Oldmans Creek
Christina River
Salem Canal
NEW CASTLE
Salem Creek
Chesapeake Delaware Canal
SALEM
Liston Point-Mile 48.3
NEW JERSEY
DELAWARE
Smyrna River
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
DELAWARE ESTUARY
COMPREHENSIVE STUDY
SECTIONS FOR
MATHEMATICAL MODEL
SCALE MILES
5 0 5 10
U. S. ENVIRONMENTAL PROTECTION AGENCY
Region II
New York, New York

FIGURE 8.

TABLE 2

POINT SOURCE NITROGEN LOADINGS TO THE DELAWARE ESTUARY

Source	DRBC River Mile	Seg No	Organic N (#N/day)	Ammonia-N (#N/day)	Nitrite N (#N/day)	Nitrate N (#N/day)	Total N (#/day)
Del River at Trenton	134.0	1	9,000	1,000.	---	16,000	26,000
Trenton	132.0	1	740.	7,020.	10.	40.	7,810.
Philadelphia N. E.	104.2	9	6,620.	8,930.	10.	110.	15,670.
Camden	97.9	13	2,280.	2,870.	330.	700.	5,180.
Philadelphia S. E.	96.6	14	4,210.	9,230.	---	---	13,440.
Texaco	93.9	15	---	1,000.	---	---	1,000.
Philadelphia S. W.	90.7	16	4,320.	13,780.	---	---	18,100.
Mobil	87.5	17	---	3,600.	---	---	3,600.
Repauno	85.5	17	---	18,000.	---	---	18,000.
C.D.C.A.	85.0	17	1,000.	1,230.	40.	50.	2,320.
Chester	80.8	18	1,000.	1,270.	---	---	2,270.
Sinclair	80.6	18	---	2,000.	---	---	2,000.
Sun Oil	78.8	19	---	7,900.	---	---	7,099.
Wilmington	72.2	21	8,250.	5,100.	250.	800.	14,400.
Carneys Pt.	71.2	21	---	---	---	30,500.	30,500.
Chambers	69.8	22	---	---	---		

In order to reflect nitrogen inputs from natural runoff and smaller tributaries, it was assumed that 1.0 mg/l ammonia nitrogen were contributed along the length of the estuary with an associated flow of 20 CFS/miles. This is equivalent to a uniform loading of 107 lbs/day^{-1}mile^{-1} as shown on the system waste vector of the computer program.

*** NITRIFICATION MODEL FOR DELAWARE ESTUARY * 1964 CONDITIONS ***

NUMBER OF SECTIONS = 30 NUMBER OF REACTION COMPONENTS = 4

SCALE FACTORS

SCALE(1) = 1000.000 SCALE(2) = 1.000 SCALE(3) = 1.000 SCALE(4) = 1000.000

INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)	INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)	INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)
(1- 1)	6500.	0.400	-3000.0	(1- 2)	15800.	0.400	3303.0	(1- 0)	0.	0.0	0.0
(1- 0)	0.	0.0	0.0	(1- 0)	0.	0.0	0.0	(2- 0)	0.	0.0	0.0
(2- 3)	21400.	0.400	3429.0	(2- 1)	15800.	0.400	-3303.0	(2- 0)	0.	0.0	0.0
(2- 0)	0.	0.0	0.0	(2- 0)	0.	0.0	0.0	(3- 0)	0.	0.0	0.0
(3- 4)	24600.	0.400	3537.0	(3- 2)	21400.	0.400	-3429.0	(3- 0)	0.	0.0	0.0
(3- 0)	0.	0.0	0.0	(3- 0)	0.	0.0	0.0	(4- 0)	0.	0.0	0.0
(4- 5)	28500.	0.400	3547.0	(4- 3)	24600.	0.400	-3537.0	(4- 0)	0.	0.0	0.0
(4- 0)	0.	0.0	0.0	(4- 0)	0.	0.0	0.0	(5- 0)	0.	0.0	0.0
(5- 6)	34100.	0.400	3580.0	(5- 4)	28500.	0.400	-3547.0	(5- 0)	0.	0.0	0.0
(5- 0)	0.	0.0	0.0	(5- 0)	0.	0.0	0.0	(6- 0)	0.	0.0	0.0
(6- 7)	41400.	0.400	3670.0	(6- 5)	34100.	0.400	-3580.0	(6- 0)	0.	0.0	0.0
(6- 0)	0.	0.0	0.0	(6- 0)	0.	0.0	0.0	(7- 0)	0.	0.0	0.0
(7- 8)	49600.	0.900	3540.0	(7- 6)	41400.	0.400	-3670.0	(7- 0)	0.	0.0	0.0
(7- 0)	0.	0.0	0.0	(7- 0)	0.	0.0	0.0	(8- 0)	0.	0.0	0.0
(8- 9)	51200.	1.600	3565.0	(8- 7)	49600.	0.900	-3540.0	(8- 0)	0.	0.0	0.0
(8- 0)	0.	0.0	0.0	(8- 0)	0.	0.0	0.0	(9- 0)	0.	0.0	0.0
(9- 10)	55400.	2.200	3597.0	(9- 8)	51200.	1.600	-3565.0	(9- 0)	0.	0.0	0.0
(9- 0)	0.	0.0	0.0	(9- 0)	0.	0.0	0.0	(10- 0)	0.	0.0	0.0
(10- 11)	60900.	2.700	3617.0	(10- 9)	55400.	2.200	-3597.0	(10- 0)	0.	0.0	0.0
(10- 0)	0.	0.0	0.0	(10- 0)	0.	0.0	0.0	(11- 0)	0.	0.0	0.0
(11- 12)	65000.	3.300	3847.0	(11- 10)	60900.	2.700	-3617.0	(11- 0)	0.	0.0	0.0
(11- 0)	0.	0.0	0.0	(11- 0)	0.	0.0	0.0	(12- 0)	0.	0.0	0.0
(12- 13)	66000.	3.800	3893.0	(12- 11)	65000.	3.300	-3847.0	(12- 0)	0.	0.0	0.0
(12- 0)	0.	0.0	0.0	(12- 0)	0.	0.0	0.0	(13- 0)	0.	0.0	0.0
(13- 14)	72700.	4.400	3898.0	(13- 12)	66000.	3.800	-3893.0	(13- 0)	0.	0.0	0.0
(13- 0)	0.	0.0	0.0	(13- 0)	0.	0.0	0.0	(14- 0)	0.	0.0	0.0
(14- 15)	88300.	4.500	3908.0	(14- 13)	72700.	4.400	-3898.0	(14- 0)	0.	0.0	0.0
(14- 0)	0.	0.0	0.0	(14- 0)	0.	0.0	0.0	(15- 0)	0.	0.0	0.0
(15- 16)	98000.	4.600	3973.0	(15- 14)	88300.	4.500	-3908.0	(15- 0)	0.	0.0	0.0
(15- 0)	0.	0.0	0.0	(15- 0)	0.	0.0	0.0	(16- 0)	0.	0.0	0.0
(16- 17)	104900.	4.800	4682.0	(16- 15)	98000.	4.600	-3973.0	(16- 0)	0.	0.0	0.0
(16- 0)	0.	0.0	0.0	(16- 0)	0.	0.0	0.0	(17- 0)	0.	0.0	0.0
(17- 18)	113400.	5.000	5051.0	(17- 16)	104900.	4.800	-4682.0	(17- 0)	0.	0.0	0.0
(17- 0)	0.	0.0	0.0	(17- 0)	0.	0.0	0.0	(18- 0)	0.	0.0	0.0
(18- 19)	126300.	5.300	5164.0	(18- 17)	113400.	5.000	-5051.0	(18- 0)	0.	0.0	0.0
(18- 0)	0.	0.0	0.0	(18- 0)	0.	0.0	0.0	(19- 0)	0.	0.0	0.0
(19- 20)	142800.	5.500	5239.0	(19- 18)	126300.	5.300	-5164.0	(19- 0)	0.	0.0	0.0
(19- 0)	0.	0.0	0.0	(19- 0)	0.	0.0	0.0	(20- 0)	0.	0.0	0.0
(20- 21)	150400.	5.800	5271.0	(20- 19)	142800.	5.500	-5239.0	(20- 0)	0.	0.0	0.0
(20- 0)	0.	0.0	0.0	(20- 0)	0.	0.0	0.0	(21- 0)	0.	0.0	0.0
(21- 22)	151900.	6.000	5282.0	(21- 20)	150400.	5.800	-5271.0	(21- 0)	0.	0.0	0.0
(21- 0)	0.	0.0	0.0	(21- 0)	0.	0.0	0.0	(22- 0)	0.	0.0	0.0
(22- 23)	162900.	6.200	5601.0	(22- 21)	151900.	6.000	-5282.0	(22- 0)	0.	0.0	0.0
(22- 0)	0.	0.0	0.0	(22- 0)	0.	0.0	0.0	(23- 0)	0.	0.0	0.0
(23- 24)	176800.	6.300	5603.0	(23- 22)	162900.	6.200	-5601.0	(23- 0)	0.	0.0	0.0
(23- 0)	0.	0.0	0.0	(23- 0)	0.	0.0	0.0	(24- 0)	0.	0.0	0.0
(24- 25)	181700.	6.400	5605.0	(24- 23)	176800.	6.300	-5603.0	(24- 0)	0.	0.0	0.0
(24- 0)	0.	0.0	0.0	(24- 0)	0.	0.0	0.0	(25- 0)	0.	0.0	0.0
(25- 26)	188400.	6.500	5608.0	(25- 24)	181700.	6.400	-5605.0	(25- 0)	0.	0.0	0.0
(25- 0)	0.	0.0	0.0	(25- 0)	0.	0.0	0.0	(26- 0)	0.	0.0	0.0
(26- 27)	196400.	6.600	5614.0	(26- 25)	188400.	6.500	-5608.0	(26- 0)	0.	0.0	0.0
(26- 0)	0.	0.0	0.0	(26- 0)	0.	0.0	0.0	(27- 0)	0.	0.0	0.0
(27- 28)	214500.	6.800	5624.0	(27- 26)	196400.	6.600	-5614.0	(27- 0)	0.	0.0	0.0
(27- 0)	0.	0.0	0.0	(27- 0)	0.	0.0	0.0				

INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)	INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)	INTERFACE ROW-SEG	AREA (FT**2)	E (MI**2/D)	Q (CFS)
(28- 29)	235000.	6.900	5627.0	(28- 27)	214500.	6.800	-5624.0	(28- 0)	0.	0.0	0.0
(28- 0)	0.	0.0	0.0	(28- 0)	0.	0.0	0.0	(28- 0)	0.	0.0	0.0
(29- 30)	254500.	7.000	5697.0	(29- 28)	235000.	6.900	-5627.0	(29- 0)	0.	0.0	0.0
(29- 0)	0.	0.0	0.0	(29- 0)	0.	0.0	0.0	(29- 0)	0.	0.0	0.0
(30- 30)	307400.	7.200	5722.0	(30- 29)	254500.	7.000	-5697.0	(30- 0)	0.	0.0	0.0
(30- 0)	0.	0.0	0.0	(30- 0)	0.	0.0	0.0	(30- 0)	0.	0.0	0.0

SECTION 1 HAS AN EXCESS FLOW OF 0.303000E 03 CFS
SECTION 2 HAS AN EXCESS FLOW OF 0.126000E 03 CFS
SECTION 3 HAS AN EXCESS FLOW OF 0.108000E 03 CFS
SECTION 4 HAS AN EXCESS FLOW OF 0.999983E 01 CFS
SECTION 5 HAS AN EXCESS FLOW OF 0.330000E 02 CFS
SECTION 6 HAS AN EXCESS FLOW OF 0.900000E 02 CFS
SECTION 7 HAS AN EXCESS FLOW OF -.130000E 03 CFS
SECTION 8 HAS AN EXCESS FLOW OF 0.250003E 02 CFS
SECTION 9 HAS AN EXCESS FLOW OF 0.319997E 02 CFS
SECTION 10 HAS AN EXCESS FLOW OF 0.200000E 02 CFS
SECTION 11 HAS AN EXCESS FLOW OF 0.230000E 03 CFS
SECTION 12 HAS AN EXCESS FLOW OF 0.459999E 02 CFS
SECTION 13 HAS AN EXCESS FLOW OF 0.499992E 01 CFS
SECTION 14 HAS AN EXCESS FLOW OF 0.100002E 02 CFS
SECTION 15 HAS AN EXCESS FLOW OF 0.649996E 02 CFS
SECTION 16 HAS AN EXCESS FLOW OF 0.709000E 03 CFS
SECTION 17 HAS AN EXCESS FLOW OF 0.369000E 03 CFS
SECTION 18 HAS AN EXCESS FLOW OF 0.113000E 03 CFS
SECTION 19 HAS AN EXCESS FLOW OF 0.749999E 02 CFS
SECTION 20 HAS AN EXCESS FLOW OF 0.320001E 02 CFS
SECTION 21 HAS AN EXCESS FLOW OF 0.110001E 02 CFS
SECTION 22 HAS AN EXCESS FLOW OF 0.319000E 03 CFS
SECTION 23 HAS AN EXCESS FLOW OF 0.199981E 01 CFS
SECTION 24 HAS AN EXCESS FLOW OF 0.200019E 01 CFS
SECTION 25 HAS AN EXCESS FLOW OF 0.299972E 01 CFS
SECTION 26 HAS AN EXCESS FLOW OF 0.600020E 01 CFS
SECTION 27 HAS AN EXCESS FLOW OF 0.999983E 01 CFS
SECTION 28 HAS AN EXCESS FLOW OF 0.300010E 01 CFS
SECTION 29 HAS AN EXCESS FLOW OF 0.699999E 02 CFS
SECTION 30 HAS AN EXCESS FLOW OF 0.250000E 02 CFS

CHARAC. LENGTHS OF SEGMENTS (FT)

INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH	INTERFACE	LENGTH
1 - 1	21000.	1 - 2	21000.	1 - 0	0.	1 - 0	0.	1 - 0	0.	1 - 0	0.
2 - 3	20000.	2 - 1	20000.	2 - 0	0.	2 - 0	0.	2 - 0	0.	2 - 0	0.
3 - 4	20000.	3 - 2	20000.	3 - 0	0.	3 - 0	0.	3 - 0	0.	3 - 0	0.
4 - 5	20000.	4 - 3	20000.	4 - 0	0.	4 - 0	0.	4 - 0	0.	4 - 0	0.
5 - 6	20000.	5 - 4	20000.	5 - 0	0.	5 - 0	0.	5 - 0	0.	5 - 0	0.
6 - 7	20000.	6 - 5	20000.	6 - 0	0.	6 - 0	0.	6 - 0	0.	6 - 0	0.
7 - 8	10000.	7 - 6	10000.	7 - 0	0.	7 - 0	0.	7 - 0	0.	7 - 0	0.
8 - 9	10000.	8 - 7	10000.	8 - 0	0.	8 - 0	0.	8 - 0	0.	8 - 0	0.
9 - 10	10000.	9 - 8	10000.	9 - 0	0.	9 - 0	0.	9 - 0	0.	9 - 0	0.
10 - 11	10000.	10 - 9	10000.	10 - 0	0.	10 - 0	0.	10 - 0	0.	10 - 0	0.
11 - 12	10000.	11 - 10	10000.	11 - 0	0.	11 - 0	0.	11 - 0	0.	11 - 0	0.
12 - 13	10000.	12 - 11	10000.	12 - 0	0.	12 - 0	0.	12 - 0	0.	12 - 0	0.
13 - 14	10000.	13 - 12	10000.	13 - 0	0.	13 - 0	0.	13 - 0	0.	13 - 0	0.
14 - 15	10000.	14 - 13	10000.	14 - 0	0.	14 - 0	0.	14 - 0	0.	14 - 0	0.
15 - 16	20000.	15 - 14	20000.	15 - 0	0.	15 - 0	0.	15 - 0	0.	15 - 0	0.
16 - 17	20000.	16 - 15	20000.	16 - 0	0.	16 - 0	0.	16 - 0	0.	16 - 0	0.
17 - 18	20000.	17 - 16	20000.	17 - 0	0.	17 - 0	0.	17 - 0	0.	17 - 0	0.
18 - 19	20000.	18 - 17	20000.	18 - 0	0.	18 - 0	0.	18 - 0	0.	18 - 0	0.
19 - 20	20000.	19 - 18	20000.	19 - 0	0.	19 - 0	0.	19 - 0	0.	19 - 0	0.
20 - 21	20000.	20 - 19	20000.	20 - 0	0.	20 - 0	0.	20 - 0	0.	20 - 0	0.
21 - 22	10000.	21 - 20	10000.	21 - 0	0.	21 - 0	0.	21 - 0	0.	21 - 0	0.
22 - 23	10000.	22 - 21	10000.	22 - 0	0.	22 - 0	0.	22 - 0	0.	22 - 0	0.
23 - 24	10000.	23 - 22	10000.	23 - 0	0.	23 - 0	0.	23 - 0	0.	23 - 0	0.
24 - 25	10000.	24 - 23	10000.	24 - 0	0.	24 - 0	0.	24 - 0	0.	24 - 0	0.
25 - 26	10000.	25 - 24	10000.	25 - 0	0.	25 - 0	0.	25 - 0	0.	25 - 0	0.
26 - 27	10000.	26 - 25	10000.	26 - 0	0.	26 - 0	0.	26 - 0	0.	26 - 0	0.
27 - 28	10000.	27 - 26	10000.	27 - 0	0.	27 - 0	0.	27 - 0	0.	27 - 0	0.
28 - 29	10000.	28 - 27	10000.	28 - 0	0.	28 - 0	0.	28 - 0	0.	28 - 0	0.
29 - 30	20000.	29 - 28	20000.	29 - 0	0.	29 - 0	0.	29 - 0	0.	29 - 0	0.
30 - 30	20000.	30 - 29	20000.	30 - 0	0.	30 - 0	0.	30 - 0	0.	30 - 0	0.

VALUES OF ALPHA AND EPRIM

INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA	INTRFC	EPRIM (MGD)	ALPHA
1- 1	26.	0.993	1- 2	64.	0.985	1- 0	0.	0.0	1- 0	0.	0.0	1- 0	0.	0.0	1- 0	0.	0.0
2- 3	89.	0.980	2- 1	64.	0.985	2- 0	0.	0.0	2- 0	0.	0.0	2- 0	0.	0.0	2- 0	0.	0.0
3- 4	103.	0.978	3- 2	89.	0.980	3- 0	0.	0.0	3- 0	0.	0.0	3- 0	0.	0.0	3- 0	0.	0.0
4- 5	119.	0.974	4- 3	103.	0.978	4- 0	0.	0.0	4- 0	0.	0.0	4- 0	0.	0.0	4- 0	0.	0.0
5- 6	142.	0.969	5- 4	119.	0.974	5- 0	0.	0.0	5- 0	0.	0.0	5- 0	0.	0.0	5- 0	0.	0.0
6- 7	230.	0.951	6- 5	142.	0.969	6- 0	0.	0.0	6- 0	0.	0.0	6- 0	0.	0.0	6- 0	0.	0.0
7- 8	931.	0.797	7- 6	230.	0.951	7- 0	0.	0.0	7- 0	0.	0.0	7- 0	0.	0.0	7- 0	0.	0.0
8- 9	1709.	0.500	8- 7	931.	0.797	8- 0	0.	0.0	8- 0	0.	0.0	8- 0	0.	0.0	8- 0	0.	0.0
9- 10	2542.	0.500	9- 8	1709.	0.500	9- 0	0.	0.0	9- 0	0.	0.0	9- 0	0.	0.0	9- 0	0.	0.0
10- 11	3429.	0.500	10- 9	2542.	0.500	10- 0	0.	0.0	10- 0	0.	0.0	10- 0	0.	0.0	10- 0	0.	0.0
11- 12	4474.	0.500	11- 10	3429.	0.500	11- 0	0.	0.0	11- 0	0.	0.0	11- 0	0.	0.0	11- 0	0.	0.0
12- 13	5231.	0.500	12- 11	4474.	0.500	12- 0	0.	0.0	12- 0	0.	0.0	12- 0	0.	0.0	12- 0	0.	0.0
13- 14	6671.	0.500	13- 12	5231.	0.500	13- 0	0.	0.0	13- 0	0.	0.0	13- 0	0.	0.0	13- 0	0.	0.0
14- 15	5525.	0.667	14- 13	6671.	0.500	14- 0	0.	0.0	14- 0	0.	0.0	14- 0	0.	0.0	14- 0	0.	0.0
15- 16	4701.	0.500	15- 14	5525.	0.667	15- 0	0.	0.0	15- 0	0.	0.0	15- 0	0.	0.0	15- 0	0.	0.0
16- 17	5251.	0.500	16- 15	4701.	0.500	16- 0	0.	0.0	16- 0	0.	0.0	16- 0	0.	0.0	16- 0	0.	0.0
17- 18	5913.	0.500	17- 16	5251.	0.500	17- 0	0.	0.0	17- 0	0.	0.0	17- 0	0.	0.0	17- 0	0.	0.0
18- 19	6980.	0.500	18- 17	5913.	0.500	18- 0	0.	0.0	18- 0	0.	0.0	18- 0	0.	0.0	18- 0	0.	0.0
19- 20	8190.	0.500	19- 18	6980.	0.500	19- 0	0.	0.0	19- 0	0.	0.0	19- 0	0.	0.0	19- 0	0.	0.0
20- 21	12129.	0.333	20- 19	8190.	0.500	20- 0	0.	0.0	20- 0	0.	0.0	20- 0	0.	0.0	20- 0	0.	0.0
21- 22	19008.	0.500	21- 20	12129.	0.333	21- 0	0.	0.0	21- 0	0.	0.0	21- 0	0.	0.0	21- 0	0.	0.0
22- 23	21064.	0.500	22- 21	19008.	0.500	22- 0	0.	0.0	22- 0	0.	0.0	22- 0	0.	0.0	22- 0	0.	0.0
23- 24	23230.	0.500	23- 22	21064.	0.500	23- 0	0.	0.0	23- 0	0.	0.0	23- 0	0.	0.0	23- 0	0.	0.0
24- 25	24253.	0.500	24- 23	23230.	0.500	24- 0	0.	0.0	24- 0	0.	0.0	24- 0	0.	0.0	24- 0	0.	0.0
25- 26	25540.	0.500	25- 24	24253.	0.500	25- 0	0.	0.0	25- 0	0.	0.0	25- 0	0.	0.0	25- 0	0.	0.0
26- 27	27034.	0.500	26- 25	25540.	0.500	26- 0	0.	0.0	26- 0	0.	0.0	26- 0	0.	0.0	26- 0	0.	0.0
27- 28	30420.	0.500	27- 26	27034.	0.500	27- 0	0.	0.0	27- 0	0.	0.0	27- 0	0.	0.0	27- 0	0.	0.0
28- 29	22545.	0.667	28- 27	30420.	0.500	28- 0	0.	0.0	28- 0	0.	0.0	28- 0	0.	0.0	28- 0	0.	0.0
29- 30	18577.	0.500	29- 28	22545.	0.667	29- 0	0.	0.0	29- 0	0.	0.0	29- 0	0.	0.0	29- 0	0.	0.0
30- 30	23080.	0.500	30- 29	18577.	0.500	30- 0	0.	0.0	30- 0	0.	0.0	30- 0	0.	0.0	30- 0	0.	0.0

SECTION	VOLUME (MGAL)	TEMPERATURE (CENT.)	CHLORIDE CONC (MG/L)
1	1810.16	30.CC	14.
2	2722.72	30.CC	14.
3	3440.80	28.CC	15.
4	3979.36	28.CC	15.
5	4757.28	28.CC	15.
6	5654.88	27.5C	15.
7	3403.40	26.5C	16.
8	3769.92	26.CC	17.
9	3986.84	26.CC	19.
10	4353.36	26.CC	20.
11	4712.40	26.CC	21.
12	4959.40	27.CC	22.
13	5191.12	27.CC	25.
14	6021.40	27.CC	40.
15	13912.80	27.CC	51.
16	15184.40	26.5C	43.
17	16336.32	26.5C	100.
18	17922.08	26.5C	115.
19	20136.16	26.5C	235.
20	21931.36	26.CC	400.
21	11309.76	26.CC	650.
22	11773.52	26.CC	663.
23	12761.04	26.CC	1350.
24	13404.16	26.CC	1420.
25	13838.CC	26.CC	1660.
26	14291.52	26.CC	1900.
27	15363.92	26.CC	2180.
28	16815.04	26.CC	2400.
29	36622.07	26.CC	2860.
30	42037.59	26.CC	3350.

SYSTEM REACTION RATE CONSTANTS (1/DAY)

REACTION	THETA	SECTION NUMBER												
		1	2	3	4	5	6	7	8	9	10	11	12	13
		14	15	16	17	18	19	20	21	22	23	24	25	26
		27	28	29	30									
1 IC 2	0.0	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
		0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
		0.1000	0.1000	0.1000	0.1000									
2 IC 3	1.080	0.2375	0.2375	0.2036	0.2036	0.2036	0.1959	0.1814	0.1746	0.1746	0.1746	0.1746	0.1885	0.0017
		0.0017	0.0017	0.0016	0.0825	0.0825	0.1814	0.1746	0.1746	0.1746	0.1746	0.1746	0.1746	
		0.1746	0.1746	0.1746	0.1746									
4 IC 4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0500
		0.0500	0.0500	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0500	0.0500	0.0500									
3 IC 4	0.0	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.0020
		0.0020	0.0020	0.0020	0.1100	0.1100	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
		0.3000	0.3000	0.3000	0.3000									
2 IC 2	0.0	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200
		0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200
		0.1200	0.1200	0.1200	0.1200									
3 IC 3	0.0	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
		0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
		0.3000	0.3000	0.3000	0.3000									

DIRECT DISCHARGES (LB/DAY)

LOAD	SECTION	COMPONENT
10165.0	1	1
405.0	2	1
405.0	3	1
405.0	4	1
405.0	5	1
405.0	6	1
202.0	7	1
202.0	8	1
6822.0	9	1
202.0	10	1
202.0	11	1
202.0	12	1
2482.0	13	1
4412.0	14	1
405.0	15	1
4725.0	16	1
1405.0	17	1
1405.0	18	1
405.0	19	1
405.0	20	1
8452.0	21	1
202.0	22	1
202.0	23	1
202.0	24	1
202.0	25	1
202.0	26	1
202.0	27	1
202.0	28	1
405.0	29	1
405.0	30	1
8445.0	1	2
405.0	2	2
405.0	3	2
405.0	4	2
405.0	5	2
405.0	6	2
202.0	7	2
202.0	8	2
9132.0	9	2
202.0	10	2
202.0	11	2
202.0	12	2
2072.0	13	2
9432.0	14	2
1405.0	15	2
14185.0	16	2
23235.0	71	2
3675.0	18	2
8305.0	19	2
405.0	20	2
5302.0	21	2
202.0	22	2
202.0	23	2
202.0	24	2
202.0	25	2
202.0	26	2
202.0	27	2

Reproduced from best available copy.

202.0	28	2
405.0	29	2
405.0	30	2
10.0	1	3
10.0	9	3
330.0	13	3
40.0	17	3
250.0	21	3
16040.0	1	4
110.0	9	4
700.0	13	4
50.0	17	4
31300.0	21	4

SYSTEM BOUNDARY CONCENTRATIONS

SECTION	COMPONENT	BOUND. CONC. (MG/L)

SYSTEM SOURCE VECTOR

COMPONENT	SECTION	LOAD (LB/DAY)
1	1	10165.0
1	2	405.0
1	3	405.0
1	4	405.0
1	5	405.0
1	6	405.0
1	7	202.0
1	8	202.0
1	9	6822.0
1	10	202.0
1	11	202.0
1	12	202.0
1	13	2462.0
1	14	4412.0
1	15	405.0
1	16	4725.0
1	17	1405.0
1	18	1405.0
1	19	405.0
1	20	405.0
1	21	8452.0
1	22	202.0
1	23	202.0
1	24	202.0
1	25	202.0
1	26	202.0
1	27	202.0
1	28	202.0
1	29	405.0
1	30	405.0
2	1	8445.0
2	2	405.0
2	3	405.0
2	4	405.0
2	5	405.0
2	6	405.0
2	7	202.0
2	8	202.0
2	9	9132.0
2	10	202.0
2	11	202.0
2	12	202.0
2	13	2072.0
2	14	4432.0
2	15	1405.0
2	16	14185.0
2	17	0.0
2	18	3675.0
2	19	8305.0
2	20	405.0
2	21	5302.0
2	22	202.0
2	23	202.0
2	24	202.0
2	25	202.0
2	26	202.0
2	27	202.0
2	28	202.0

Reproduced from best available copy.

2	29	405.0
2	30	405.0
3	1	10.0
3	2	0.0
3	3	0.0
3	4	0.0
3	5	0.0
3	6	0.0
3	7	0.0
3	8	0.0
3	9	10.0
3	10	0.0
3	11	0.0
3	12	0.0
3	13	330.0
3	14	0.0
3	15	0.0
3	16	0.0
3	17	40.0
3	18	0.0
3	19	0.0
3	20	0.0
3	21	250.0
3	22	0.0
3	23	0.0
3	24	0.0
3	25	0.0
3	26	0.0
3	27	0.0
3	28	0.0
3	29	0.0
3	30	0.0
4	1	16040.0
4	2	0.0
4	3	0.0
4	4	0.0
4	5	0.0
4	6	0.0
4	7	0.0
4	8	0.0
4	9	110.0
4	10	0.0
4	11	23235.0
4	12	0.0
4	13	700.0
4	14	0.0
4	15	0.0
4	16	0.0
4	17	50.0
4	18	0.0
4	19	0.0
4	20	0.0
4	21	31300.0
4	22	0.0
4	23	0.0
4	24	0.0
4	25	0.0
4	26	0.0
4	27	0.0
4	28	0.0
4	29	0.0

0.0
3C
4

SYSTEM SOLUTION VECTOR

SECTION	MILE PT	COMPONENT CONCENTRATION (MG/L) 1	2	3	4
1	129.51	0.568	0.472	0.077	0.915
2	125.72	0.569	0.476	0.155	0.940
3	121.93	0.573	0.482	0.206	1.006
4	118.14	0.592	0.501	0.251	1.135
5	114.35	0.608	0.515	0.287	1.303
6	110.56	0.615	0.522	0.305	1.495
7	108.67	0.659	0.562	0.326	1.727
8	106.78	0.721	0.632	0.346	1.939
9	104.89	1.007	0.979	0.423	2.339
10	103.00	1.021	0.970	0.457	2.826
11	101.11	1.039	0.977	0.454	3.376
12	99.22	1.104	1.045	0.419	3.253
13	97.33	1.201	1.101	0.286	2.946
14	95.44	1.264	1.275	0.198	2.705
15	91.65	1.257	1.261	0.125	2.473
16	87.86	1.246	1.303	0.114	2.518
17	84.07	1.228	1.205	0.240	2.896
18	80.28	1.230	1.199	0.346	3.554
19	76.49	1.217	1.195	0.519	4.410
20	72.70	1.202	1.112	0.563	5.098
21	70.81	1.185	1.057	0.553	5.424
22	68.92	1.113	0.983	0.532	5.350
23	67.03	1.047	0.917	0.507	5.230
24	65.14	0.975	0.848	0.476	5.012
25	63.25	0.894	0.774	0.439	4.684
26	61.36	0.803	0.693	0.395	4.247
27	59.50	0.705	0.605	0.347	3.707
28	57.58	0.604	0.518	0.298	3.106
29	53.79	0.462	0.386	0.223	2.251
30	50.00	0.223	0.190	0.109	1.082

Reproduced from best available copy.

DISSOLVED OXYGEN DEFICIT ANALYSIS

SEGMENT	TEMPERATURE (CENT.)	REAERATION RATE (1/DAY)
1	30.00	0.228175
2	30.00	0.228175
3	28.00	0.217605
4	28.00	0.217605
5	28.00	0.217605
6	27.50	0.215040
7	26.50	0.210000
8	26.00	0.207525
9	26.00	0.207525
10	26.00	0.207525
11	26.00	0.207525
12	27.00	0.212505
13	27.00	0.212505
14	27.00	0.212505
15	27.00	0.212505
16	26.50	0.210000
17	26.50	0.210000
18	26.50	0.210000
19	26.50	0.210000
20	26.00	0.207525
21	26.00	0.207525
22	26.00	0.207525
23	26.00	0.207525
24	26.00	0.207525
25	26.00	0.207525
26	26.00	0.207525
27	26.00	0.207525
28	26.00	0.207525
29	26.00	0.207525
30	26.00	0.207525

DEFICIT DUE TO REACTION 2 TO 3 STOCHIOMETRIC COEFFICIENT OF 3.43

SECTION	MILE PT	DEFICIT CONCENTRATION (MG/L)	D.O. SATURATION (MG/L)	DISSOLVED OXYGEN (MG/L)
1	129.5	0.275	7.436	7.161
2	125.7	0.579	7.436	6.857
3	121.9	0.805	7.722	6.917
4	118.1	1.023	7.722	6.699
5	114.4	1.210	7.722	6.512
6	110.6	1.335	7.796	6.461
7	108.7	1.465	7.944	6.480
8	106.8	1.588	8.020	6.432
9	104.9	1.916	8.020	6.104
10	103.0	2.078	8.020	5.942
11	101.1	2.068	8.020	5.952
12	99.2	1.922	7.869	5.948
13	97.3	1.396	7.869	6.473
14	95.4	1.044	7.868	6.824
15	91.6	0.724	7.867	7.143
16	87.9	0.657	7.942	7.286
17	84.1	1.159	7.938	6.780
18	80.3	1.645	7.937	6.292
19	76.5	2.380	7.929	5.549
20	72.7	2.621	7.992	5.372
21	70.8	2.596	7.974	5.379
22	68.9	2.516	7.973	5.457
23	67.0	2.417	7.924	5.507

24	65.1	2.283	7.919	5.636
25	63.3	2.113	7.901	5.788
26	61.4	1.913	7.884	5.971
27	59.5	1.685	7.864	6.178
28	57.6	1.450	7.848	6.398
29	53.8	1.086	7.815	6.728
30	50.0	0.535	7.779	7.244

DEFICIT DUE TO REACTION 3 TO 4 STOICHIOMETRIC COEFFICIENT OF 1.14

SECTION	MILE PT	DEFICIT CONCENTRATION (MG/L)	D.O. SATURATION (MG/L)	DISSOLVED OXYGEN (MG/L)
1	129.5	0.019	7.436	7.417
2	125.7	0.066	7.436	7.371
3	121.9	0.128	7.722	7.594
4	118.1	0.201	7.722	7.521
5	114.4	0.277	7.722	7.445
6	110.6	0.344	7.796	7.452
7	108.7	0.403	7.944	7.541
8	106.8	0.449	8.020	7.571
9	104.9	0.516	8.020	7.504
10	103.0	0.553	8.020	7.466
11	101.1	0.538	8.020	7.481
12	99.2	0.478	7.869	7.391
13	97.3	0.343	7.869	7.526
14	95.4	0.251	7.868	7.616
15	91.6	0.164	7.867	7.703
16	87.9	0.124	7.942	7.819
17	84.1	0.178	7.938	7.760
18	80.3	0.302	7.937	7.635
19	76.5	0.550	7.929	7.379
20	72.7	0.691	7.992	7.301
21	70.8	0.725	7.974	7.249
22	68.9	0.726	7.973	7.248
23	67.0	0.715	7.924	7.209
24	65.1	0.689	7.919	7.230
25	63.3	0.648	7.901	7.253
26	61.4	0.594	7.884	7.290
27	59.5	0.529	7.864	7.335
28	57.6	0.456	7.848	7.390
29	53.8	0.347	7.815	7.468
30	50.0	0.172	7.779	7.608

TOTAL DISSOLVED OXYGEN DEFICIT FOR REACTIONS
2 TO 3
3 TO 4

SECTION	MILE PT	TOTAL DEFICIT CONCENTRATION (MG/L)	DISSOLVED OXYGEN (MG/L)
1	129.5	0.295	7.142
2	125.7	0.645	6.791
3	121.9	0.933	6.789
4	118.1	1.224	6.498
5	114.4	1.487	6.235
6	110.6	1.679	6.117
7	108.7	1.868	6.077
8	106.8	2.037	5.983
9	104.9	2.432	5.588
10	103.0	2.632	5.388
11	101.1	2.666	5.413
12	99.2	2.400	5.470
13	97.3	1.739	6.130
14	95.4	1.295	6.573
15	91.6	0.889	6.978
16	87.9	0.780	7.162
17	84.1	1.337	6.601
18	80.3	1.947	5.990
19	76.5	2.430	4.999
20	72.7	3.312	4.681
21	70.8	3.320	4.654
22	68.9	3.242	4.732
23	67.0	3.132	4.792
24	65.1	2.972	4.947
25	63.3	2.761	5.140
26	61.4	2.507	5.377
27	59.5	2.214	5.649
28	57.6	1.908	5.940
29	53.8	1.433	6.382
30	50.0	0.707	7.073

Reproduced from best available copy.

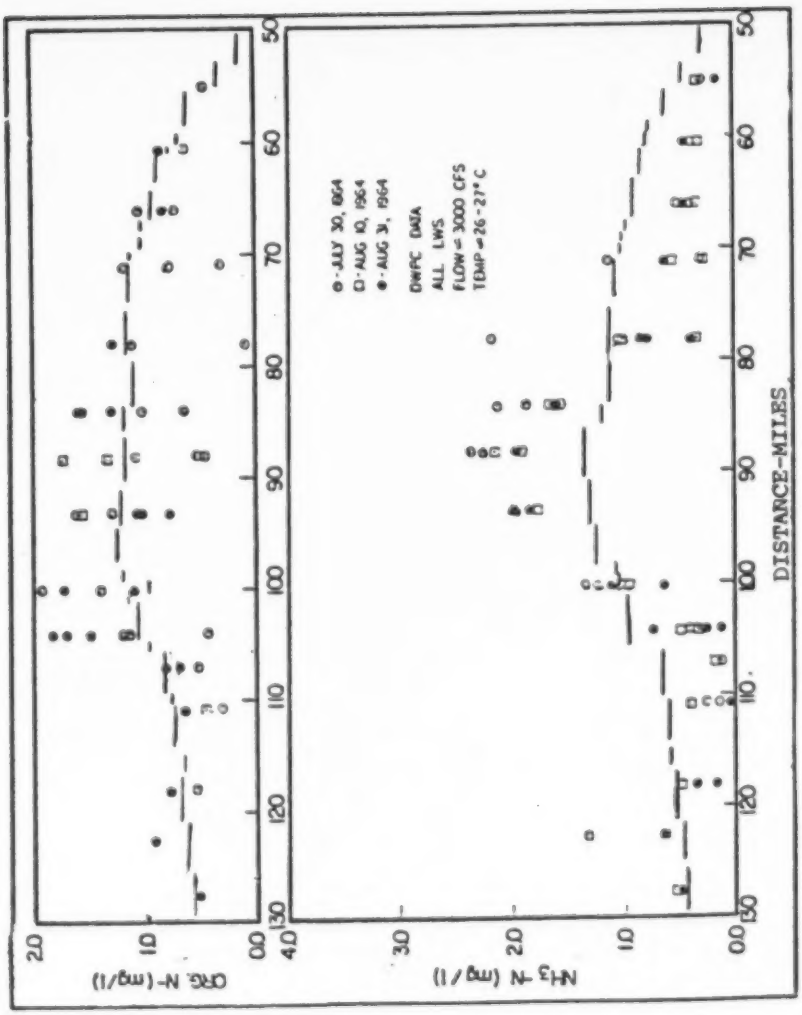


FIGURE 9: Observed vs Computed (Dashed Line) Nutrient Profiles, August 1964

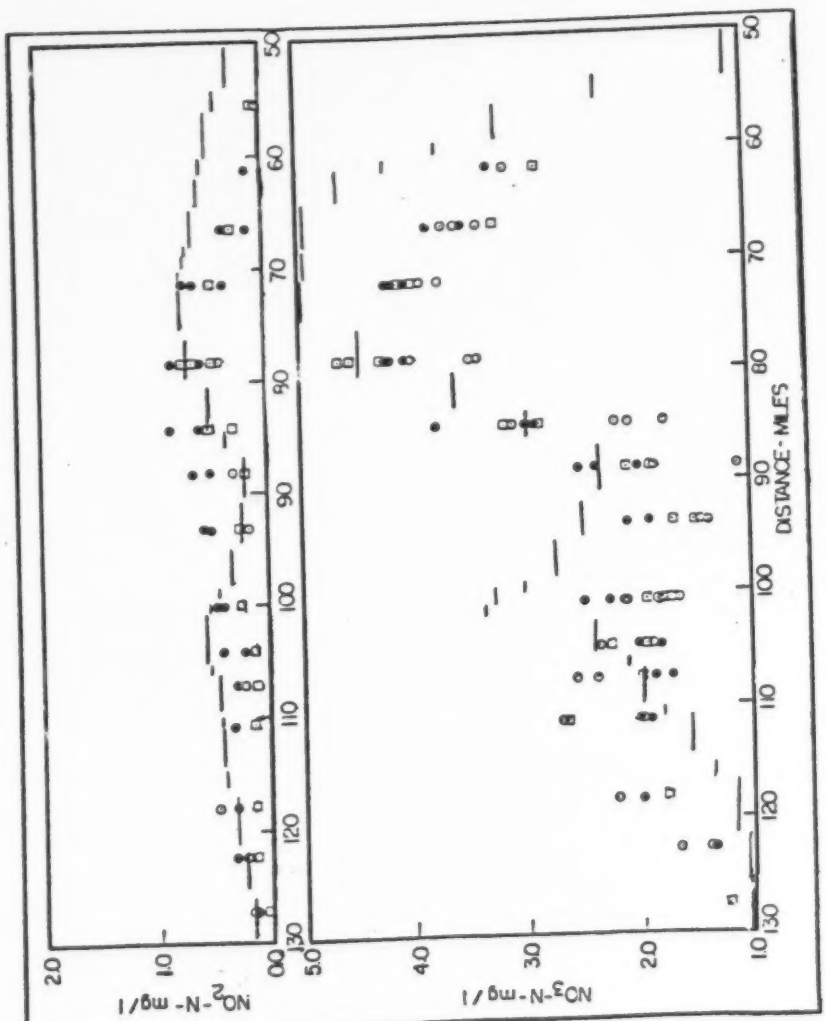


FIGURE 10: Observed vs Computed (Dashed Line) Nutrient Profiles, August 1964

The following output pages are a second method of computing deficit profiles for the Delaware Estuary, where the computation of dissolved oxygen deficit is accomplished by defining a deficit "species" as part of the system.

In this case it is defined as the N_5 component. Since ammonia and nitrite (N_2 and N_3 correspondingly) are the species producing deficit, a reaction scheme for these two species is defined. The reaction rates for each of these is the product of the stochiometric coefficient for the oxydation reaction times the reaction rate constant of the species that consumes oxygen.

It should be noted that this method computes total deficit and not the contribution of each component. For a large system, this technique significantly increases the amount of computer time and main memory required, since it increases the order of the various matrices by the number of segments in the system.

The pages that follow begin with the "System Reaction Rate Constants", since the preceding pages would be identical to the example shown previously.

SYSTEM REACTION RATE CONSTANTS (1/DAY)

REACTION	THETA	SECTION NUMBER 1	2	3	4	5	6	7	8	9	10	11	12	13
		14	15	16	17	18	19	20	21	22	23	24	25	26
		27	28	29	30									
1 TO 2	0.0	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
		0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
		0.1000	0.1000	0.1000	0.1000									
2 TO 3	1.080	0.2375	0.2375	0.2036	0.2036	0.2036	0.1959	0.1814	0.1746	0.1746	0.1746	0.1746	0.1885	0.0017
		0.0017	0.0017	0.0016	0.0825	0.0825	0.1814	0.1746	0.1746	0.1746	0.1746	0.1746	0.1746	0.1746
		0.1746	0.1746	0.1746	0.1746									
4 TO 4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0500
		0.0500	0.0500	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0500	0.0500	0.0500									
2 TO 5	1.080	0.8146	0.8146	0.6984	0.6984	0.6984	0.6720	0.6222	0.5987	0.5987	0.5987	0.5987	0.6466	0.0058
		0.0058	0.0058	0.0056	0.2828	0.2828	0.6222	0.5987	0.5987	0.5987	0.5987	0.5987	0.5987	0.5987
		0.5987	0.5987	0.5987	0.5987									
3 TO 5	0.0	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.0020
		0.0020	0.0020	0.0020	0.1250	0.1250	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420	0.3420
		0.3420	0.3420	0.3420	0.3420									
5 TO 5	0.0	0.2282	0.2282	0.2176	0.2176	0.2176	0.2150	0.2100	0.2075	0.2075	0.2075	0.2075	0.2125	0.2125
		0.2125	0.2125	0.2100	0.2100	0.2100	0.2100	0.2075	0.2075	0.2075	0.2075	0.2075	0.2075	0.2075
		0.2075	0.2075	0.2075	0.2075									
3 TO 4	0.0	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.0020
		0.0020	0.0020	0.0020	0.1100	0.1100	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
		0.3000	0.3000	0.3000	0.3000									
2 TO 2	0.0	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200
		0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200	0.1200
		0.1200	0.1200	0.1200	0.1200									
3 TO 3	0.0	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
		0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
		0.3000	0.3000	0.3000	0.3000									

DIRECT DISCHARGES (LB/DAY)

LOAD	SECTION	COMPONENT
10165.0	1	1
405.0	2	1
405.0	3	1
405.0	4	1
405.0	5	1
405.0	6	1
202.0	7	1
202.0	8	1
6822.0	9	1
202.0	10	1
202.0	11	1
202.0	12	1
2482.0	13	1
4412.0	14	1
405.0	15	1
4725.0	16	1
1405.0	17	1
1405.0	18	1
405.0	19	1
405.0	20	1
8452.0	21	1
202.0	22	1
202.0	23	1
202.0	24	1
202.0	25	1
202.0	26	1
202.0	27	1
202.0	28	1
405.0	29	1
405.0	30	1
8445.0	1	2
405.0	2	2
405.0	3	2
405.0	4	2
405.0	5	2
405.0	6	2
202.0	7	2
202.0	8	2
9132.0	9	2
202.0	10	2
202.0	11	2
202.0	12	2
2072.0	13	2
9432.0	14	2
1405.0	15	2
14185.0	16	2
23235.0	71	2
3675.0	18	2
8305.0	19	2
405.0	20	2
5302.0	21	2
202.0	22	2
202.0	23	2
202.0	24	2
202.0	25	2
202.0	26	2
202.0	27	2

202.0	28	2
405.0	29	2
405.0	30	2
10.0	1	3
10.0	9	3
330.0	13	3
40.0	17	3
250.0	21	3
16040.0	1	4
110.0	9	4
700.0	13	4
50.0	17	4
31300.0	21	4

SYSTEM BOUNDARY CONCENTRATIONS

SECTION	COMPONENT	BOUND. CONC. (MG/L)

SYSTEM SOURCE VECTOR

COMPONENT	SECTION	LOAD (LB/DAY)
1	1	10165.0
1	2	405.0
1	3	405.0
1	4	405.0
1	5	405.0
1	6	405.0
1	7	202.0
1	8	202.0
1	9	6822.0
1	10	202.0
1	11	202.0
1	12	202.0
1	13	2482.0
1	14	4412.0
1	15	405.0
1	16	4725.0
1	17	1405.0
1	18	1405.0
1	19	405.0
1	20	405.0
1	21	8452.0
1	22	202.0
1	23	202.0
1	24	202.0
1	25	202.0
1	26	202.0
1	27	202.0
1	28	202.0
1	29	405.0
1	30	405.0
2	1	8445.0
2	2	405.0
2	3	405.0
2	4	405.0
2	5	405.0
2	6	405.0
2	7	202.0
2	8	202.0
2	9	9132.0
2	10	202.0
2	11	202.0
2	12	202.0
2	13	2072.0
2	14	9432.0
2	15	1405.0
2	16	14185.0
2	17	0.0
2	18	3675.0
2	19	8305.0
2	20	405.0
2	21	5302.0
2	22	202.0
2	23	202.0
2	24	202.0
2	25	202.0
2	26	202.0
2	27	202.0
2	28	202.0

	2	29	405.0
	2	30	405.0
	3	1	10.0
	3	2	0.0
	3	3	0.0
	3	4	0.0
	3	5	0.0
	3	6	0.0
	3	7	0.0
	3	8	0.0
	3	9	10.0
	3	10	0.0
	3	11	0.0
	3	12	0.0
	3	13	330.0
	3	14	0.0
	3	15	0.0
	3	16	0.0
	3	17	40.0
	3	18	0.0
	3	19	0.0
	3	20	0.0
	3	21	250.0
	3	22	0.0
	3	23	0.0
	3	24	0.0
	3	25	0.0
	3	26	0.0
	3	27	0.0
	3	28	0.0
	3	29	0.0
	3	30	0.0
	4	1	16040.0
	4	2	0.0
	4	3	0.0
	4	4	0.0
	4	5	0.0
	4	6	0.0
	4	7	0.0
	4	8	0.0
	4	9	110.0
	4	10	0.0
	4	11	23235.0
	4	12	0.0
	4	13	700.0
	4	14	0.0
	4	15	0.0
	4	16	0.0
	4	17	50.0
	4	18	0.0
	4	19	0.0
12	4	20	0.0
11	4	21	31300.0
10	4	22	0.0
9	4	23	0.0
8	4	24	0.0
7	4	25	0.0
6	4	26	0.0
5	4	27	0.0
4	4	28	0.0
3	4	29	0.0

14113

4	30	0.0
5	1	0.0
5	2	0.0
5	3	0.0
5	4	0.0
5	5	0.0
5	6	0.0
5	7	0.0
5	8	0.0
5	9	0.0
5	10	0.0
5	11	0.0
5	12	0.0
5	13	0.0
5	14	0.0
5	15	0.0
5	16	0.0
5	17	0.0
5	18	0.0
5	19	0.0
5	20	0.0
5	21	0.0
5	22	0.0
5	23	0.0
5	24	0.0
5	25	0.0
5	26	0.0
5	27	0.0
5	28	0.0
5	29	0.0
5	30	0.0

SYSTEM SOLUTION VECTOR

SECTION	MILE PT	COMPONENT CONCENTRATION (MG/L) 1	2	3	4	5
1	129.51	0.568	0.472	0.077	0.915	0.295
2	125.72	0.569	0.476	0.155	0.940	0.645
3	121.93	0.573	0.482	0.206	1.006	0.933
4	118.14	0.592	0.501	0.251	1.135	1.224
5	114.35	0.608	0.515	0.287	1.303	1.487
6	110.56	0.615	0.522	0.305	1.495	1.679
7	108.67	0.659	0.562	0.326	1.727	1.868
8	106.78	0.721	0.632	0.348	1.939	2.037
9	104.89	1.007	0.979	0.423	2.339	2.432
10	103.00	1.021	0.970	0.457	2.826	2.632
11	101.11	1.039	0.977	0.454	3.376	2.607
12	99.22	1.104	1.045	0.419	3.253	2.400
13	97.33	1.202	1.161	0.286	2.946	1.739
14	95.44	1.264	1.275	0.198	2.705	1.295
15	91.65	1.257	1.261	0.125	2.473	0.888
16	87.86	1.246	1.303	0.114	2.518	0.780
17	84.07	1.228	1.205	0.240	2.896	1.336
18	80.28	1.230	1.199	0.346	3.554	1.947
19	76.49	1.217	1.195	0.519	4.410	2.930
20	72.70	1.202	1.112	0.563	5.098	3.312
21	70.81	1.185	1.057	0.553	5.424	3.321
22	68.92	1.113	0.983	0.532	5.350	3.242
23	67.03	1.047	0.917	0.507	5.230	3.132
24	65.14	0.975	0.848	0.476	5.012	2.972
25	63.25	0.894	0.774	0.439	4.684	2.762
26	61.36	0.803	0.693	0.395	4.247	2.507
27	59.50	0.705	0.605	0.347	3.707	2.215
28	57.58	0.604	0.518	0.298	3.106	1.908
29	53.79	0.452	0.386	0.223	2.251	1.433
30	50.00	0.223	0.190	0.109	1.082	0.707

Execution at the WCC facility

The two versions of the program have been stored as core-image modules on a partitioned dataset at the EPA Washington Computer Center Facility (COMNET). Users of this facility may execute the programs by using the JCL stream listed below.

```
//EPAIII   JOB ('AAAA',MIII,,'FEDBAK03',B,,20),TIME=(5,30)
//JOBLIB DD UNIT=3330-1,VOL=SER=USER66,DISP=OLD,
//  DSNAME=CN.EPAGAN.R2TW.WATER.MODELS
//S1  EXEC PGM=PPPPPPPP,REGION=300K
//GO.FT01F001 DD UNIT=SYSDA,DSNAME=+FILE1,DISP=(NEW DELETE),
//  SPACE=(720,(720,5)),DCB=(RECFM=FS,BLKSIZE=720,LRECL=720)
//GO.FT03F001 DD UNIT=SYSDA,DSNAME=+FILE3,DISP=(NEW,DELETE),
//  SPACE=(1280,(400,5)),DCB=(RECFM=FS,BLKSIZE=1280,LRECL=1280)
//GO.FT06F001 DD SYSOUT=A
//GO.FT07F001 DD SYSOUT=B
//GO.FT09F001 DD SYSOUT=A,DCB=BLKSIZE=120
//GO.FT08F001 DD *
      ** DATA CARDS GO HERE **
/*
//
```

WHERE III = USER'S INITIALS

AAAA = ACCOUNT NUMBER

PPPPPPPP=FEDBAK3A FOR 120 SEGMENT MODEL ...

PPPPPPPP=FEDBAK3B FOR 180 SEGMENT MODEL

ACKNOWLEDGEMENTS

Some material presented in this report is based on extensions of the HAR03[4] computer model, originally documented by Steve Chapra.

The typing was very professionally handled by Ms. Gloria Kasporwitz.

REFERENCES FOR THE COMPUTER PROGRAM

(1) Committee on Sanitary Engineering Research, Solubility of Atmospheric Oxygen in Water, J. Sanit. Eng. Div., Am. Soc. Civil Engrs., SA 4, 41 (1960)

(2) Nitrification of the Delaware Estuary, prepared for the Delaware River Basin Commission by Hydroscience, Inc., Westwood, N.J., June 1969.

(3) Thomann, R.V., Systems Analysis and Water Quality Management, Environmental Science Division, New York, 1971.

(4) Chapra, S.C. and Nossa G.A. HARM3 A Computer Program for the Modelling of Water Quality Parameters in a Steady State Multi dimensional Aquatic System., USEPA Region II, New York, N.Y., October 1974.

APPENDIX A

LISTING OF INPUT DATA FOR VERIFICATION OF DELAWARE ESTUARY

```
 *** NITRIFICATION MODEL FOR DELAWARE ESTUARY *  1964 CONDITIONS *** W/ DEFICIT
3004
     1
 1000.  1.0      1.0  1000.
  6.5   .40 -3000.  1 15.8  .40  3303.   2

  21.4  .40  3429.  3

  24.6  .40  3537.  4

  28.5  .40  3547.  5

  34.1  .40  3580.  6

  41.4  .40  3670.  7

  49.6  .90  3540.  8

  51.2  1.6  3565.  9

  55.4  2.2  3597. 10

  60.9  2.7  3617. 11

  65.0  3.3  3847. 12

  66.0  3.8  3893. 13

  72.7  4.4  3898. 14

 88.3   4.5  3908. 15

 98.0   4.6  3973. 16

104.9   4.8  4682. 17

113.4   5.0  5051. 18

126.3   5.3  5164. 19

142.8   5.5  5239. 20

150.4   5.8  5271. 21

151.9   6.0  5282. 22

162.9   6.2  5601. 23

176.8   6.3  5603. 24

181.7   6.4  5605. 25

188.4   6.5  5608. 26

196.4   6.6  5614. 27

214.5   6.8  5624. 28

235.0   6.9  5627. 29
```

```
254.5   7.0  5697. 30

307.4   7.2  5722. 30

  1 21.        2  21.
  3 20.        1  20.
  4 20.        2  20.
  5 20.        3  20.
  6 20.        4  20.
  7 20.        5  20.
  8 10.        6  10.
  9 10.        7  10.
 10 10.        8  10.
 11 10.        9  10.
 12 10.       10  10.
 13 10.       11  10.
 14 10.       12  10.
 15 10.       13  10.
 16 20.       14  20.
 17 20.       15  20.
 18 20.       16  20.
 19 20.       17  20.
 20 20.       18  20.
 21 20.       19  20.
 22 10.       20  10.
 23 10.       21  10.
 24 10.       22  10.
 25 10.       23  10.
 26 10.       24  10.
 27 10.       25  10.
 28 10.       26  10.
 29 10.       27  10.
 30 20.       28  20.
 30 20.       29  20.
   14.         14.      15.      15.      15.      15.      16.      17.
   19.         20.      21.      22.      25.      40.      51.      43.
  100.        115.     235.     400.     650.     663.    1350.    1420.
 1660.       1900.     2180.    2400.    2860.    3350.
   30.        242.     3 .      364.      28.     460.      28.     532.
   28.        636.     27.5     756.     26.5     455.      26.     504.
   26.        533.     26.      582.     26.      630.      27.     655.
   27.        694.     27.      805.     27.     1860.     26.5    2030.
 26.5        2184.     26.5    2396.     26.5    2692.      26.    2932.
   26.        1512.    26.      1574.    26.     1698.      26.    1792.
   26.        1850.    26.      1924.    26.     2054.      26.    2248.
   26.        4896.    26.      5620.
0102
  .10          .10      .10      .10      .10      .10      .10      .10
  .10          .10      .10      .10      .10      .10      .10      .10
  .10          .10      .10      .10      .10      .10      .10      .10
  .10          .10      .10      .10      .10      .10
0203 1.08
  .11          .11      .11      .11      .11      .11      .11      .11
  .11          .11      .11      .11      .001     .001     .001     .001
  .05          .05      .11      .11      .11      .11      .11      .11
  .11          .11      .11      .11      .11      .11
0404
    0.0          0.0      0.0      0.0      0.0      0.0      0.0      0.0
```

```
    0.0       0.0       0.0       0.0       .05       .05       .05       0.0
    0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
    0.0       0.0       0.0       .05       .05       .05
0304
   .30       .30       .30       .30       .30       .30       .30       .30
   .30       .30       .30       .30       .002      .002      .002      .002
  .110       .110      .30       .30       .30       .30       .30       .30
   .30       .30       .30       .30       .30       .30
0202
    .12       .12       .12       .12       .12       .12       .12       .12
    .12       .12       .12       .12       .12       .12       .12       .12
    .12       .12       .12       .12       .12       .12       .12       .12
    .12       .12       .12       .12       .12       .12
0303
    .30       .30       .30       .30       .30       .30       .30       .30
    .30       .30       .30       .30       .30       .30       .30       .30
    .30       .30       .30       .30       .30       .30       .30       .30
    .30       .30       .30       .30       .30       .30

 10165.     1 1
  405.      2 1
  405.      3 1
  405.      4 1
  405.      5 1
  405.      6 1
  202.      7 1
  202.      8 1
 6822.      9 1
  202.     10 1
  202.     11 1
  202.     12 1
 2482.     13 1
 4412.     14 1
  405.     15 1
 4725.     16 1
 1405.     17 1
 1405.     18 1
  405.     19 1
  405.     20 1
 8452.     21 1
  202.     22 1
  202.     23 1
  202.     24 1
  202.     25 1
  202.     26 1
  202.     27 1
  202.     28 1
  405.     29 1
  405.     30 1
 8445.      1 2
  405.      2 2
  405.      3 2
  405.      4 2
  405.      5 2
  405.      6 2
  202.      7 2
  202.      8 2
 9132.      9 2
  202.     10 2
```

```
  202.     11 2
  202.     12 2
 2072.     13 2
 9432.     14 2
 1405.     15 2
14185.     16 2
23235.     71 2
 3675.     18 2
 8305.     19 2
  405.     20 2
 5302.     21 2
  202.     22 2
  202.     23 2
  202.     24 2
202.       25 2
  202.     26 2
  202.     27 2
  202.     28 2
  405.     29 2
  405.     30 2
   10.      1 3
   10.      9 3
  330.     13 3
   40.     17 3
  250.     21 3
16040.      1 4
  110.      9 4
  700.     13 4
   50.     17 4
31300.     21 4

 129.51     125.72     121.93   118.14   114.35   110.56   108.67   106.78
 104.89     103.0      101.11    99.22    97.33    95.44    91.65    87.86
  84.07      80.28      76.49    72.70    70.81    68.92    67.03    65.14
  63.25      61.36      59.50    57.58    53.79    50.0
.18

    2       2 3  3.43 3 4 1.14

/*
//
```